



Universitat de Lleida

# TREBALL FINAL DE GRAU



ESCOLA  
POLITÀCNICA SUPERIOR  
UNIVERSITAT DE LLEIDA  
INSPIRING THE FUTURE

**Estudiant:** MARCOS RAMÓN PEULA

**Titulació:** Grau en Enginyeria Electrònica Industrial i Automàtica

**Títol de Treball Final de Grau:**

Desenvolupament d'un curs d'aprenentatge en línia d'introducció a  
l'electrònica digital mitjançant l'ús de FPGAs lliures

**Director/a:** ALBERT SAIZ VELA / CONCEPCIÓ ROIG MATEU

Presentació

Mes: Juliol

Any: 2020

## **AGRAÏMENTS**

Aquest Treball de Fi de Grau no hagués estat possible sense la col·laboració de moltes persones a les que vull expressar el meu agraïment.

En primer lloc, al meu tutor del TFG, el professor Albert Saiz Vela, per la seva inestimable guia i ajuda des del primer dia i especialment durant l'anòmala situació produïda per la pandèmia de COVID19, enmig de la qual s'ha dut a terme aquest projecte. També a Concepció Roig Mateu, que amb molt encert em va posar en contacte amb ell.

Igualment a tots els professors de la UdL, que al llarg d'aquests anys han estat involucrats en la meva formació acadèmica del Grau en Enginyeria Electrònica Industrial i Automàtica.

Vull també agrair als companys amb els quals hem fet junts el trajecte tots aquests anys d'estudiants.

I finalment a la meua família pel seu suport incondicional, en tot moment i com sempre.

## Índex:

Índex:.....	3
1. INTRODUCCIÓ.....	6
1.1 La importància de la pràctica de laboratori en els estudis d'enginyeria.....	6
1.2 Metodologia actual: avantatges i inconvenients .....	6
1.2.1. Esquema actual de la realització de les pràctiques.....	8
1.2.2. Avantatges i inconvenients .....	9
1.3. Objectius .....	11
1.5 Metodologia: .....	12
2. PLAQUES FPGA .....	14
2.1 Què són les FPGAs? .....	14
2.2 Arquitectura de les FPGAs.....	15
2.2.1 Cel·les Lògiques/Elements Lògics:.....	16
2.2.2 Slice: .....	16
2.2.3 CLBs i LABs.....	17
2.2.4 Taules de recerca (LUT) .....	18
2.2.5 Elements d'emmagatzemament: Flip-Flops.....	19
2.2.6 Lògica d'acarreig.....	20
2.2.7 Generació i Distribució de la Senyal de Rellotge.....	20
2.2.8 Blocs d'E/S (IOB) .....	21
2.2.9 Bancs d'E/S:.....	22
2.2.10 Buffer d'E/S: .....	23
2.2.11 Interconnexions a les FPGAs: .....	24
2.3 Futur de les FPGA: .....	25
2.4 Fabricants de FPGAs.....	25
2.4.1 Lattice Semiconductors .....	26
2.4.2 INTEL.....	27
2.4.3 Xilinx .....	29
2.4.4 Microsemi.....	30
2.5 Llenguatge de Programació.....	32
2.5.1 VHDL.....	33
2.5.2 Verilog .....	35
3. HARDWARE LLIURE /SOFTWARE LLIURE .....	38
3.1 Concepte .....	38
3.2 Avantatges i inconvenients d'eines SF/HW lliures .....	38

3.3. Antecedents. El món Maker .....	38
3.4. El projecte ICESTORM .....	39
3.5. Plaques de la família Ice40: .....	41
4. PLACA ICEZUM ALHAMBRA II .....	44
4.1 Característiques.....	45
4.2 Connectors .....	46
4.4 Connectors d' E/S mascle de 5V amb alimentació i massa .....	47
4.5 Bill of Material .....	48
4.6 Schemàtics.....	49
4.7 PCB .....	51
5. TRAINING BOARD DE LA UdL.....	52
5.1 Objectius .....	52
5.2 Característiques.....	52
5.3 Schematic: .....	53
5.4 PCB: .....	56
6. CURS D'APRENENTATGE EN LÍNIA.....	57
6.1 Github.....	57
6.2 Pràctiques proposades.....	58
6.2.1 Pràctica 1 .....	58
6.2.2 Pràctica 2 .....	67
6.2.3 Pràctica 3 .....	73
6.2.4 Pràctica 4 .....	81
7 . TinyFPGA .....	93
7.1 Schematic .....	94
7.2 PCB .....	95
8. CONCLUSIÓ.....	96
9. REFERÈNCIES BIBLIOGRÀFIQUES.....	97

## **Resum**

L'estudi dels circuits electrònics digitals en els graus que incorporen assignatures d'electrònica digital comporta la realització de pràctiques de laboratori. Aquestes pràctiques es feien tradicionalment mitjançant l'ús de circuits integrats monolítics, que presentaven problemes associats al muntatge. Recentment a la Universitat de Lleida (UdL) s'ha implementat una nova metodologia d'ensenyament basada en l'ús de dispositius FPGAs, mitjançant la creació d'una plataforma de desenvolupament de baix cost utilitzant Hardware i Software lliure.

Aquest treball té com a propòsit complementar aquesta plataforma través la creació d'un curs d'aprenentatge en línia d'introducció a l'electrònica digital mitjançant l'ús de FPGAs lliures, amb l'objectiu que els estudiants puguin experimentar realitzant les pràctiques que s'han desenvolupat de manera més autònoma i creativa.

**PARAULES CLAU :** Circuits electrònics digitals, FPGAs, Software i Hardware lliure

## **Abstract**

The study of digital electronic circuits in degrees that incorporate assignments of digital electronics involves the performance of laboratory practices. These practices were traditionally done through the use of monolithic integrated circuits, which presented problem problems associated with assembly.

Recently at the University of Lleida (UdL) it has been implemented a new design methodology based on the FPGAs device, through the creation of a low-cost development platform using free Hardware and Software.

This work aims to complement this platform with the creation of an online learning course of introduction to digital electronics through the use of free FPGAs, with the aim that students can experiment by doing the internships. which have been developed in a more autonomous and creative way.

**KEYWORDS:** Digital electronic circuits, FPGAs, Free Software and Hardware

# **1. INTRODUCCIÓ**

## **1.1 La importància de la pràctica de laboratori en els estudis d'enginyeria**

A la titulació del grau d'Enginyeria Electrònica Industrial i Automàtica i afins, les pràctiques de laboratori tenen gran importància en la formació i el desenvolupament de les competències que els estudiants han d'adquirir. Mitjançant el paradigma “learning by doing” la pràctica en enginyeria és el que permet a l'estudiant comprovar que els conceptes que li han estat ensenyats han estat entesos. Per això és necessari que des de la Universitat s'incentivi un aprenentatge on la formació pràctica tingui un paper preponderant.

El pla d'estudis del grau de GEEIA integra diferents assignatures associades a l'electrònica digital. L'objectiu d'aquestes assignatures és transmetre a l'estudiant els coneixements necessaris que el capaciti per a dissenyar sistemes electrònics digitals complexos.

Els programes de les assignatures d'electrònica digital incorporen pràctiques de laboratori que consisteixen en la resolució d'exercicis que permeten desenvolupar les habilitades apreses, relacionades amb els continguts de l'assignatura, fent servir el material electrònic i la instrumentació disponibles.

Però cal que aquestes pràctiques no siguin activitats repetitives o repartició de tasques entre un grup d'estudiants, sinó que és necessari que l'estudiant posi a prova l'observació del fenomen i el sentit crític que li permeti ser capaç de resoldre els problemes.

Tanmateix, en els graus d'enginyeria electrònica és molt important que des de la Universitat es potenciï, a través de les pràctiques, el contacte dels estudiants amb les novetats tecnològiques i projectes de recerca capdavaners relacionats amb la microelectrònica i els sistemes digitals.

## **1.2 Metodologia actual: avantatges i inconvenients**

A les universitats des de fa anys, a l'estudi dels circuits electrònics es fan servir eines de simulació de circuits, en les quals els estudiants se centren fonamentalment en executar en aquests simuladors els circuits a estudiar i analitzar els resultats.

L'objectiu és que l'alumne es familiaritzi amb l'ús d'aquesta mena, posant en pràctica les tècniques bàsiques de disseny en electrònica digital.

Així doncs, el simulador, una aplicació software, s'ha convertit en una eina bàsica en l'aprenentatge que permet a l'alumne capturar circuits i simular el funcionament de manera virtual per a verificar que el comportament de totes les seves funcions compleix amb els requisits esperats, podent corregir els possibles errors amb anterioritat a la seva implementació física, amb el consegüent estalvi de temps i costos econòmics.

Al mercat hi ha nombroses eines de simulació, orientades al món industrial o a l'educatiu, cadascuna amb els seus avantatges i inconvenients en relació amb les seves prestacions, corba d'aprenentatge i cost. A l'àmbit de la Universitat l'elecció del software normalment es fa tenint en compte el seu cost.

A l'Escola Politècnica Superior d'Enginyeria (EPSI) de la UdL tradicionalment les pràctiques a les assignatures amb contingut d'electrònica digital s'han fet mitjançant la simulació mitjançant l'ús de simuladors lògics i la seva posterior implementació física al laboratori on es verifica el resultat.

Concretament, a la nostra Universitat es fan pràctiques de resolució de circuits lògics amb el simulador ISIS de Proteus. Proteus és un software de disseny per a PCB desenvolupat per Labcenter Electrònics que permet realitzar totes les tasques de disseny de circuits electrònics, com ara disseny de l'esquema electrònic, programació del software, construcció de la placa de circuit imprès, simulació interactiva de circuits analògics digitals i amb microcontroladors en temps real, depuració d'errors, documentació i construcció. Compta amb una ampla llibreria de components genèrics i específics.

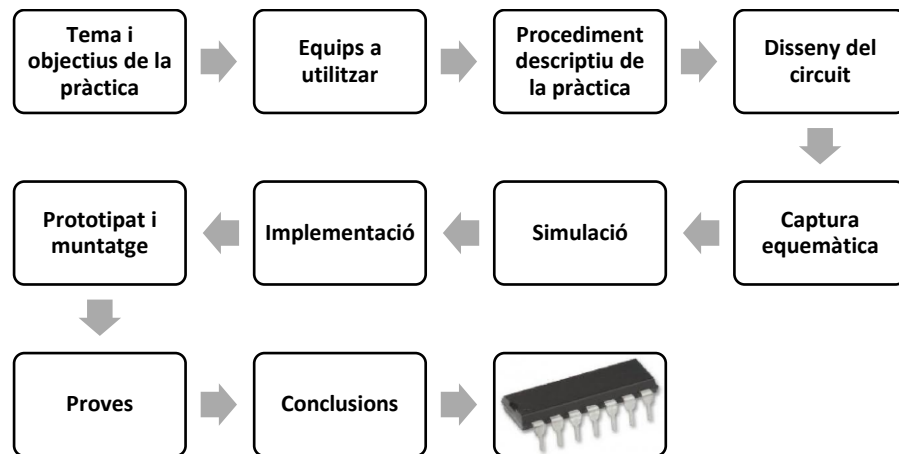
L'aplicació Proteus es compon de dos programes principals: Isis, que es fa servir per al disseny de l'esquema electrònic, i Ares, que serveix per al disseny de la placa de circuit imprès a partir de l'esquema electrònic realitzat anteriorment. També consta dels mòduls VSM que s'utilitza per a la simulació del circuit electrònic i ProSPICE, per a la simulació de la lògica del programa carregat en el microprocessador.



**Figura 1:** Proteus

Després de la simulació amb Proteus, la posterior implementació física dels circuits digitals al laboratori de la Universitat es realitza des de fa anys mitjançant la utilització de circuits integrats lògics comercials com els de la sèrie TTL 7400 o els de la sèrie CMOS 4000, cables de connexió i de l'ús de plaques tipus protoboards.

### 1.2.1. Esquema actual de la realització de les pràctiques



**Figura 2:** Diagrama de les pràctiques

Tal com es descriu en el diagrama anterior, el procés de les pràctiques segueix el següent procediment:

- 1- Es planteja el tema de la pràctica en funció del tema i conceptes tractats en classe, així com els objectius que han de ser aconseguits amb la pràctica.
- 2- Es detalla el procediment a desenvolupar, els equips, materials i dispositius que els estudiants han d'emprar per a realitzar la pràctica.
- 3- Es realitza el disseny del circuit que s'ha d'implantar en el laboratori.
- 4- Es fan els càlculs i s'executa la simulació mitjançant Proteus.
- 5- Es realitza el mapatge tecnològic i traçat de les pistes.
- 6- Disseny del circuit imprès.
- 7- Implementació física al laboratori.
- 8- Procedir a la verificació del funcionament del circuit muntat, realitzant-se les proves necessàries per a comprovar que el funcionament dels circuits desenvolupats és l'especificat.
- 9- Conclusions i observacions de la pràctica realitzada.



### 1.2.2. Avantatges i inconvenients

L'ús d'aquesta metodologia d'ensenyament té els següents avantatges:

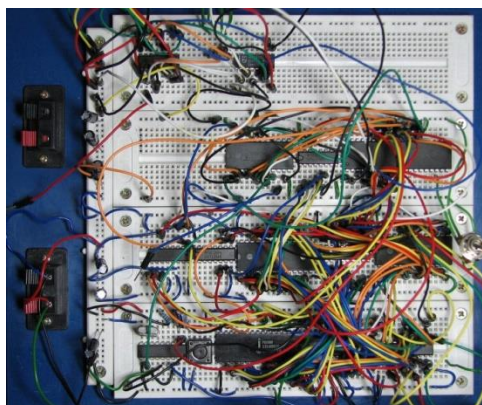
- L'ús del simulator Proteus és una eina útil per a la comprensió dels principis més bàsics del funcionament de l'electrònica digital. Dona suport a l'aprenentatge de tipus experimental, ja que l'alumne "aprèn fent".
- Amb l'ús de Proteus, s'augmenta l'interès dels alumnes, ja que li permet experimentar el que està fent com si fos real. Pot fer accions com ara moure un potenciòmetre, accionar un teclat, encendre un LED, escoltar un altaveu o fer girar un motor.
- Té un cost econòmic baix. Durant la implementació física del circuit dissenyat i la seva verificació, l'ús de circuits integrals comercials i les plaques de prototipat permet fer de manera econòmica més practiques i més variades i complexes, amb l'estalvi d'importants recursos, com és el temps i la reutilització de materials.
- Afavoreix la realització de proves sense risc. Quan un circuit treballa correctament en el simulador serà més fàcil muntar-lo en una protoboard amb la seguretat que funcionarà bé. Es pot treballar de forma més relaxada sense por de no espatllar els circuits reals, i per això es pot arriscar més.
- S'involucra de manera dinàmica als estudiants. És una metodologia que ja coneixen perquè la porten utilitzant en cursos inferiors. Són els estudiants els que poden dissenyar el seu propi circuit i es desperta el interès en muntar els elements i dispositius amb la finalitat d'observar el fenomen.
- L'aprenentatge dels processos de muntatge i testatge és molt ràpid, podent verificar-se de manera pràctica els coneixements teòrics adquirits en classe. Conèixer com estan composts els circuits i les múltiples connexions.
- Permet als alumnes descobrir qualsevol problema de disseny i corregir els seus errors.
- Els alumnes, en acabar la pràctica han pogut completar el disseny i implementació física d'un sistema digital (combinacional i seqüencial).
- Així mateix, es possibilita al professor tenir preparats diferents supòsits per a les pràctiques, sense haver de construir-les cada vegada, i coneixent per endavant els resultats.

No obstant, aquesta metodologia també té alguns inconvenients:

- Quant a l'ús del simulador Proteus, els principals inconvenients se centren en l'aspecte privatiu del software ja que es depèn de l'empresa propietària i de les

restriccions del sistema marcades per la llicència. Així mateix és necessari invertir temps per a conèixer a fons l'aplicació si es vol obtenir el major rendiment possible.

- D'altra banda, durant la implementació pràctica en el laboratori, quan el sistema digital és complex o necessita més de cinc circuits integrats comercials, el nombre de cables necessaris és molt elevat. Això fa que calgui dedicar molt de temps per a realitzar el muntatge i a causa del gran nombre de connexions manuals sovint es produeixen errors difícils de detectar per l'alumne, podent arribar a no funcionar.
- A l'haver de dedicar molt de temps al muntatge es perd temps per debatre i valorar amb el professor altres qüestions relatives al disseny i funcionament del circuit.
- Un altre inconvenient de fer les pràctiques de manera tradicional al laboratori és que es tendeixen a repetir pràctiques fixes que ja han estat fetes prèviament pel professor, i llavors, si es fa bé, no sol fallar res. I si no, com ja es coneixen els resultats, s'acaba compartint amb altres companys i provant fins que el resultat és l'esperat. Algunes vegades en pràctiques compartides no es veu tot el procés, sinó solament la part concreta que li ha correspost a l'alumne.
- Quan s'acaba la pràctica els estudiants deuen fer un informe en el qual s'indiquen totes les dades que s'han anat obtenint: construir gràfics, taules i obtenir conclusions. No obstant aquestes conclusions es basen en els objectius prèviament assenyalats pel professor, sense que hi hagi hagut probabilitat d'experimentar res a part.
- Es dona per fet que els estudiants aprenen si fan coses, però això no significa que hagin entès els processos i puguin relacionar la pràctica amb el coneixement teòric. Les pràctiques tradicionals estan molt dirigides pel professor. El paradigma "learning by doing" no té significat si no permet la creativitat.



**Figura 3** Pràctiques amb Protoboard

### 1.3. Objectius

Els problemes assenyalats en la implementació pràctica dels circuits digitals podrien millorar-se mitjançant la implementació dels circuits dissenyats i simulats a través del hardware reconfigurable de les FPGAs, que no necessita utilitzar circuits integrats comercials ni cables externs per a connectar els diferents blocs lògics.

El progressiu abaratiment dels circuits FPGAs (Field Programmable Gate Array), conjuntament amb la disponibilitat d'eines software gratuïtes en alguns casos d'accés lliure per a treballar amb aquesta mena de hardware ha obert noves possibilitats per a l'aprenentatge de l'electrònica digital.

Per tal motiu, des de la EPS-UdL s'està impulsant un canvi de metodologia en l'aprenentatge i en les pràctiques de les assignatures dels graus d'enginyeria amb continguts d'electrònica digital, mitjançant un projecte d'innovació docent, dirigit cap a un model basat en l'ús de hardware reconfigurable, com a complement als mitjans emprats actualment.

El projecte s'enfoca en el desenvolupament d'una plataforma de baix cost que inclou els elements auxiliars necessaris per a la implementació de circuits digitals complexos mitjançant l'ús de plaques d'entrenament amb FPGAs lliure.

Per a això s'ha dissenyat una training board que es compon d'una placa d'entrenament de fabricació pròpia que es connecta amb una placa FPGA, la placa Iczum Alhambra II, que actua com a nucli de la plataforma i que està preparada per al desenvolupament de circuits amb l'editor visual d'esquemes Icestudio.

En el marc d'aquest projecte d'innovació docent, aquest Treball de Fi de Grau (TFG) té com a objectiu preparar les pràctiques que realitzaran els futurs alumnes del Grau de Enginyeria Electrònica Industrial i Automàtica i se centra en el desenvolupament d'un Curs d'Aprenentatge en Línia d'Introducció a l'Electrònica Digital, seguint la filosofia del codi obert (software i hardware lliure)

Mitjançant aquesta plataforma els estudiants disposaran d'una eina d'aprenentatge pròpia, a través de la qual es possibilita una participació més activa, podent augmentar la varietat d'activitats ja que alumne pot realitzar les seves pràctiques de forma més autònoma i creativa, de manera que sigui ell mateix el que resolgui les dificultats, orientat per la informació que podrà obtenir, sense dependre del laboratori i sense haver d'invertir tant de temps en el muntatge dels circuits amb xips discrets, al mateix temps que el professor pot realitzar el seguiment d'aquestes pràctiques, guiant i donant suport durant l'aprenentatge.

Tot això pot ser complementari al sistema de pràctiques que es realitza actualment, la qual cosa facilita als alumnes tenir diferents visions i així avançar en el coneixement i funcionament dels circuits i tot això amb una infraestructura i un cost assumible.

Els objectius que es plantegen aconseguir amb aquest treball són els següents:

- Desenvolupar una eina d'aprenentatge a través d'un curs online basat en la participació activa dels alumnes, fomentant l'aprenentatge autònom.
- Realitzar d'una forma més eficaç i dinàmica les pràctiques d'electrònica digital.
- Possibilitar que els estudiants disposin de tota la informació relacionada amb la plataforma de baix cost desenvolupada.
- Obtenir una reducció en el cost de l'instrumental necessari.

### 1.5 Metodologia:

El procés de la creació del curs d'aprenentatge en línia perquè els alumnes puguin realitzar les pràctiques s'ha planificat i desenvolupat en les següents parts.

S'ha creat en **GitHub** un repositori obert a través del qual els alumnes poden accedir a la totalitat de la informació i documentació generada (esquemes, firmware, fitxers gerber de fabricació PCB, Bill of materials etc...)

Tanmateix en Github s'ha creat una wiki on estarien disponibles els arxius amb les 5 pràctiques desenvolupades prèviament. A cada pràctica s'especifica quins són els objectius que es volen aconseguir, el plantejament del problema i el procediment que s'ha de desenvolupar, incloent vídeos de funcionament dels circuits dissenyats.

Al GitHub totes les pràctiques tenen enllaços i connexió amb la training board de la UdL, tenint en compte que tant la placa training como el layout estan fets amb software lliure de manera que els estudiants la poden utilitzar, es poden descarregar els exercicis, materials, exemples , software i tot el que sigui necessari per realitzar les pràctiques, podent arribar a fabricar-se la seva pròpia plataforma de desenvolupament que els permetrà practicar fora de laboratori.

Una de les premisses de la plataforma ha estat reduir al màxim els costos, de manera que obtinguéssim com a resultat una eina de baix cost que estigués a l'abast de qualsevol. Per això, tot el material generat és de lliure utilització i està disponible perquè

qualsevol persona el pugui utilitzar, seguint la filosofia del codi obert que caracteritza el moviment maker.

Aquest TFG s'estructura de la següent forma:

Al capítol 2 es fa una introducció a las FPGAs. Què són. Les seves característiques generals. Qui són els fabricants, aplicacions programació i llenguatges.

Al capítol 3 es tractarà sobre el hardware y software lliure, avantatges i inconvenients.

Al capítol 4 s'explica la training board desenvolupada a la UdL. Les seves característiques i funcionament.

Al capítol 5 es fa l'explicació de tota la part pràctica d'aquest TFG: organització de la wiki i practiques desenvolupades.

Al capítol 6 es fa l'adaptació de la training board amb l'objectiu de substituir la placa Icezum Alhambra II per la placa TinyFPGA.

## **2. PLAQUES FPGA**

### **2.1 Què són les FPGAs?**

Les Field Programmable Gate Array (FPGAs) són dispositius semiconductors col·locats al voltant d'una matriu formada amb blocs lògics configurables (CLB) que estan connectats amb interconnexions que són programables.

La principal característica que tenen les FPGAs és que poden ser reprogramades.

El terme reprogramable o reconfigurable és el nom que se li dona a la capacitat d'una FPGA per poder implementar noves funcions després de la seva fabricació. Això es pot aconseguir gràcies a la tecnologia de programació subjacent, que pot provocar canvis en els comportaments dels xips un cop ja ha estat fabricada.

La FPGA és el més pròxim a dissenyar el propi xip des de zero, ja que permet dissenyar i implementar qualsevol funció o circuit digital dins del xip universal. Aquesta és la principal característica per la que destaca la FPGA, que en el moment de comprar-la no té cap funció definida.

Les dos principals diferències entre una FPGA i un microprocessador són que el microprocessador és un dispositiu molt complex que porta predefinida una llista d'instruccions on cada instrucció va associada a un hardware específic que està connectat de forma fixa, obligant al programador a haver de treballar únicament amb les que venen definides i configurades pel fabricant. L'altra diferència és que el microprocessador només pot executar les comandes de forma seqüencial o en sèrie.

La FPGA pot executar-les simultàniament, però això provoca que la limitació per executar múltiples comandes a la vegada sigui l'espai. Si volem realitzar moltes comandes alhora necessitarem augmentar els circuits que componen el xip i més espai per poder albergar-los.

A causa de les seves característiques tècniques, els costos i temps de desenvolupament són molt menors. És per això que la utilització de FPGAs s'està estenent en tots els àmbits industrials, com poden ser la electrònica de consum, processament de dades, automòbils, aeroespacial, defensa o indústries mèdiques.

Entre totes aquestes indústries, cal destacar la indústria de les telecomunicacions, on a causa de l'aparició dels serveis 3G,4G,LTE i 5G provocarà l'augment de la demanda de telèfons intel·ligents i l'augment de la velocitat de les connexions d'internet a les regions.

Com a conseqüència, es preveu que les FPGAs i la tecnologia SRAM tinguin una gran importància i una taxa de creixement elevada a causa de l'augment de la demanda i la seva naturalesa que permet reconfigurar els circuits a mesura que la naturalesa del disseny va variant per poder satisfer les necessitats que vagin arribant. A més, les FPGAs tenen un consum d'energia molt Baix.

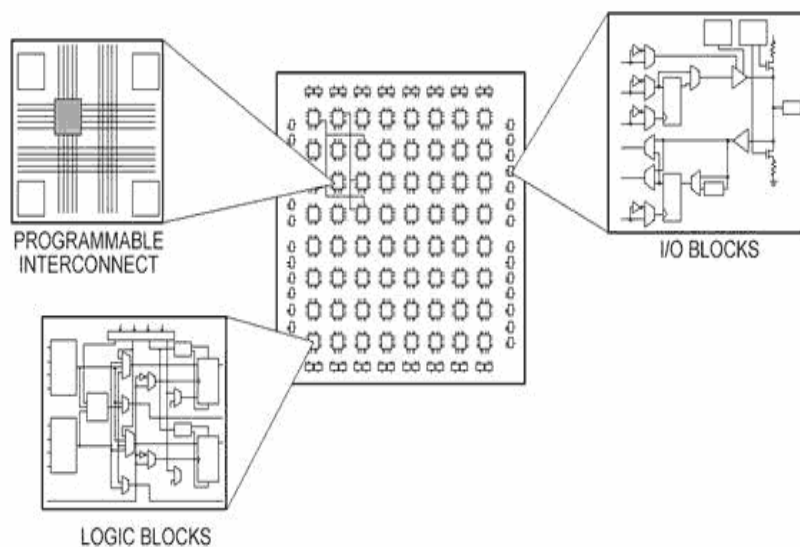
## 2.2 Arquitectura de les FPGAs

L'arquitectura d'una FPGA no està gens estandarditzada però encara que cada fabricant intenta diferenciar-se dels altres, molts dels components que utilitzen són en comú a totes les FPGAs, com són els blocs lògics configurables, blocs de memòria de doble port (dual port), blocs de control del rellotge, blocs per a execució de MACs.

(Multiplicador, Acumulador i blocs DSP), blocs E/S, etc

Es distingeixen tres elements bàsics:

- Blocs lògics programables
- Mòduls d'interconnexió programable
- Bloc d'entrada i sortides (IOB)



**Figura 4:** FPGA estructura bàsica

Una FPGA està formada per una matriu de blocs lògics configurables (CLB) que alhora està formada per Slices i aquests últims estan formats per cel·les lògiques.

### 2.2.1 Cel·les Lògiques/Elements Lògics:

La unitat més petita d'una FPGA és la cel·la lògica o element lògic segons els fabricants Xilinx o Altera.

Una cel·la lògica està formada principalment per una LUT de 4 entrades (que pot funcionar com una RAM de 16x1 o un registre de desplaçament de 16 bits), un multiplexor i un registre.

Aquest registre es pot configurar com un flip-flop (actiu per flanc) o com un latch (actiu per nivell).

### 2.2.2 Slice:

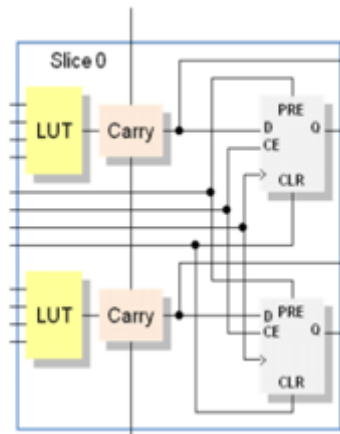
Una slice està formada per 2 o més celes lògiques individuals que comparteixen la mateixa senyal de rellotge.

Una Slice formada per 2 celes lògiques estaria formada per:

- Dos LUT
- Dos flip-flops
- Quatre sortides, dos combinacionals i dos registres
- Entrades de control per als flip-flops
- Entrades per a les LUTs
- Entrades i sortides per a la cadena d'acarreig

Dins d'un CLB hi ha dos tipus de Slices:

- SLICEM: A més de les funcions lògiques que es poden implementar en una Slice, aquesta ofereix opcions per implementar petites memòries.
- SLICEL: Només permet implementar funcions lògiques.



**Figura 5** Vista simplificada d'una SLICEL d'una FPGA Spartan



### 2.2.3 CLBs i LABs

Per sobre de les slice es troben els CLBs (Blocs Logics Configurables), com anomena el fabricant Xilinx i el LAB (Bloque de arreglo lógico), segons Altera.

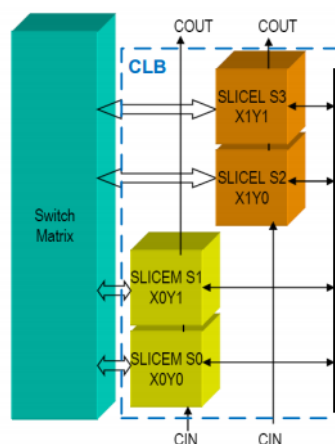
Algunes FPGAs de Xilinx tenen 2 Slices a cada CLB mentre que altres en tenen 4. La raó per tenir aquest tipus d'arquitectura jeràrquica és que és complementada per una arquitectura equivalent a la connexió.

Així doncs, s'aconsegueixen connexions molt ràpides entre els elements lògics dins de la Slice, després connexions més lentes entre les Slices dins d'un CLB i connexions lentes entre CLBs.

La finalitat és aconseguir la compensació òptima entre fer que les connexions siguin fàcils sense incórrer molt en els retards d'interconnexió.

La seva estructura macroscòpica es troba detallada a la següent imatge, on es pot apreciar un CLB constituït per:

- SLICEM S0
- SLICEM S1
- SLICEM S2
- SLICEM S3
- Rutes de connexió pel acarregament matemàtic
- Carry In i Carry Out
- Connexions a la Matriu de Connexions (Switch Matrix)
- Connexions entre Slices



**Figura 6 CLB**

Tal i com s'aprecia a la imatge anterior, aquest CLB està format per 4 slices. Cada Slice es distingeix per les seves coordenades, essent SLICE S0 el més baix a l'esquerra i SLICE S3 el més alt a la dreta.

Tant els Slices como els CLBs estan identificats per coordenades.

#### **2.2.4 Taules de recerca (LUT)**

Dins d'una FPGA, tota la seva lògica combinacional està basada en la utilització de les taules de recerca (LUT), també denominades algunes vegades Generadors de Funció. Les LUT emmagatzemen la funció lògica que el dissenyador vol implementar.

Únicament els valors de la Z són els que s'emmagatzemen dins de la LUT de 16x1.

Tot aquest procés per emmagatzemar els valors es fa mitjançant el software del fabricant, per això el procés és totalment transparent al dissenyador del sistema digital.

Una característica de molta utilitat és que el retard a través de la LUT és constant per a qualsevol funció lògica que es vulgui implementar, essent d'especial importància per a sistemes de molt alta freqüència.

Per a implementar funcions lògiques de més de quatre entrades en una LUT s'utilitzen multiplexors, especialment dedicats que estan distribuïts dins de la SLICE i del CLB per a poder implementar qualsevol funció lògica amb un nombre major de 4 entrades.

Per exemple, per a poder implementar una funció lògica amb 5 entrades, s'utilitza el multiplexor F5Mux, que multiplexarà les sortides de les dos LUTs i la cinquena entrada vindrà donada pel propi F5Mux, que funcionarà com a senyal de selecció.

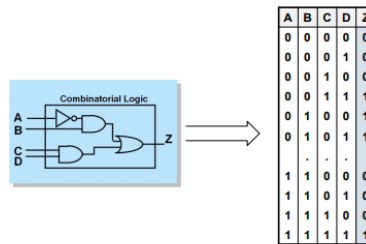
Aquest multiplexors podrien estar implementats dins de les pròpies LUTs, però al ser aquests multiplexors dedicats per aquesta funció, són més eficients i deixen a les LUTs lliures per a altres funcions.

Cada Slice disposa d'un F5Mux i d'un FiMux, on aquest segon multiplexor pot funcionar com F6Mux o F7Mux o F8Mux, segons a la ubicació en la que es trobi.

Les connexions entre aquests multiplexors es realitzen de tal forma que el retard de la senyal sigui 0.

Cal afegir que, l'última versió de la sèrie Spartan Virtex, les LUTs tenen 6 entrades. No obstant tot l'explicat anteriorment, s'aplica d'igual forma a les LUTs de 6 entrades.

La següent imatge ens mostra la semblança entre una taula de la veritat i una LUT.



**Figura 7** LUT

### 2.2.5 Elements d'emmagatzemament: Flip-Flops

Cada Slice posseeix dos elements d'emmagatzemament programables que poden funcionar tant com a flip-flop de tipus D com a latch.

L'element d'emmagatzemament ubicat a la part superior de l'Slice s'anomena FFY i el de la part inferior, FFX.

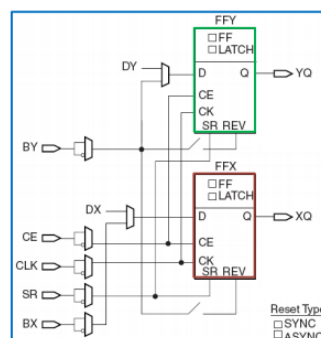
Els dos elements posseeixen un multiplexor de selecció per a la entrada D del flip-flop, on es pot seleccionar entre la sortida de la LUT respectiva.

També es pot observar que els senyals de control, com el senyal de reset o el del rellotge són comuns als dos flip-flops.

El flip-flop té construït en la seva lògica interna la opció de poder posar a zero el flip-flop de manera síncrona o asíncrona.

L'entrada REV s'utilitza per poder invertir el valor lògic d'activació del reset.

La següent imatge ens mostra les possibles configuracions dels elements d'emmagatzemament i els respectius senyals de control.



**Figura 8** Configuracions dels flip-flops d'emmagatzemament

## 2.2.6 Lògica d'acarreig

El CLB té la seva pròpia lògica dedicada exclusivament per l'acarreig que es produeix amb la suma aritmètica amb l'objectiu de poder millorar el rendiment dels sumadors, comptadors, comparadors i funcions lògiques que estiguin relacionades.

La lògica d'acarreig disponible dins d'un CLB està constituïda pels següents elements:

- Comportes lògiques dedicades exclusivament a l'acarreig
- Multiplexors dedicats
- Traçat de les pistes i connexions dedicades

## 2.2.7 Generació i Distribució de la Senyal de Rellotge

Les FPGA tenen uns blocs de lògica dedicats exclusivament a la funció de generació i control i manteniment dels senyals de rellotge.

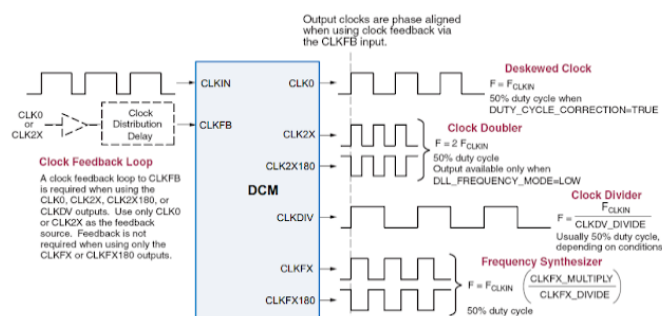
En el cas de les FPGAs produïdes pel fabricant Xilinx, aquests blocs són denominats Digital Clock Managers (DCMs).

La quantitat de DCMs que pot haver-hi incorporats dins d'una FPGA depèn en gran mesura de la grandària de la pròpia FPGA.

Segons la grandària, els DCMs d'una FPGA poden variar des de 2 DCMs per les FPGA més petites fins a 12 DCMs les més grans.

Les principals funcions de les que s'encarreguen els DCMs són:

- Eliminar el retard del senyal de rellotge (clock skew), ja sigui dins de la pròpia FPGA o amb els components externs.
- Produir corrent de fase d'un senyal de rellotge (Phase shifting), ja sigui per una fracció de període o amb increments fixes.
- Multiplicar o dividir la freqüència d'entrada generant una completament nova.
- Amplificar un senyal de rellotge.

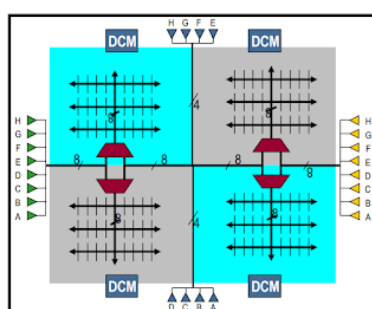


**Figura 9** Rellotge

Per a la distribució del senyal de rellotge s'utilitzen rutes de connexió que estan dedicades a aquest propòsit. Les rutes de rellotge són línies dedicades de molt baix retard, de quasi nul·la distorsió del cicle de treball i amb una oscil·lació mínima que arriben a tots els elements sincrònics de la FPGA.

La distribució d'aquestes rutes de connexió s'explica a la següent imatge, on es pot observar com la FPGA està dividida en quatre quadrants, els quals poden tenir com a màxim 8 rellotges.

Uns multiplexors dedicats seleccionaran els senyals de rellotge que s'utilitzarà en aquell quadrant en específic.



**Figura 10** Distribució de la senyal de rellotge

### 2.2.8 Blocs d'E/S (IOB)

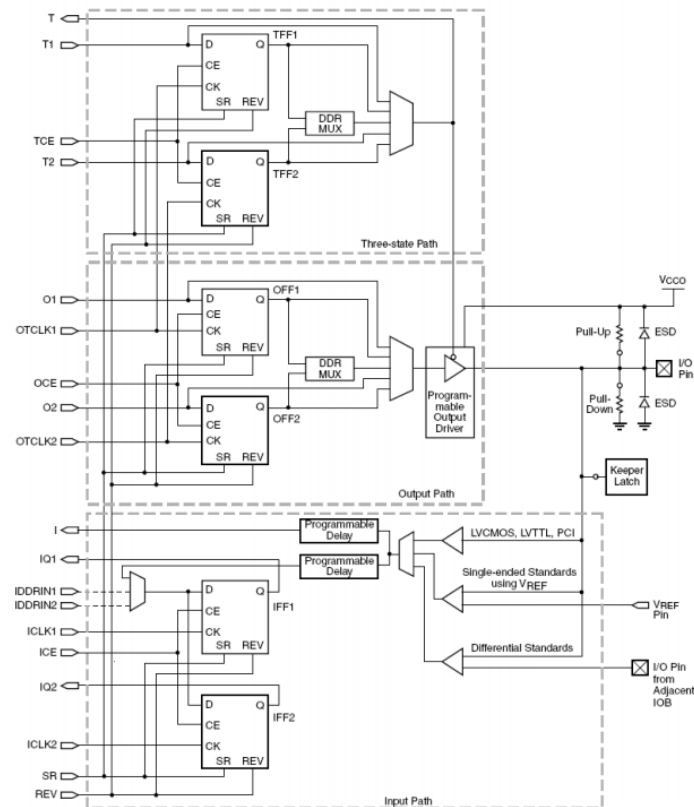
Per poder rebre i transmetre senyals digitals, les FPGAs han de disposar de blocs d'E/S.

Aquests blocs, en les FPGAs, són bastant elaborats, cosa que permet utilitzar-los amb diversos rangs de tensions i freqüències de treball, que els converteix en uns dispositius molt adaptables a les necessitats del sistema del que formen part.

Per cada terminal d'E/S existeix un bloc d'E/S, de forma que cada terminal pot ésser configurat com a entrada, sortida o bidireccional.

A cada bloc existeix un buffer que té diverses funcions configurables pel dissenyador, que permeten adaptar la FPGA en un sistema molt més complex treballant amb diverses tensions i corrents.

Una altra característica important dels IOBs és que permet el control digital de la impedància d'entrada o de sortida dels terminals del IOB.



**Figura 11** Estructura interna d'un IOB

En la imatge anterior es pot observar l'estructura interna d'un bloc d'E/S on hi ha tres possibles camins que pot recórrer el senyal digital:

- Camí d'entrada
- Camí de sortida
- Camí d'alta impedància

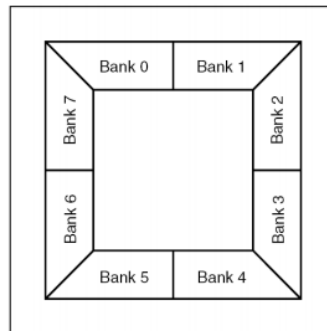
### 2.2.9 Bancs d'E/S:

Els diferents blocs d'E/S dins d'una FPGA estan agrupats en els denominats bancs d'E/S. Segons el tipus d'FPGA amb la que tractem, trobarem bancs d'entre 20 i 40 blocs d'E/S.

Cada banc té el seu propi voltatge d'alimentació (VCCO) i el seu propi voltatge de referència (Vref), que són comuns a tots els blocs.

És per aquesta disposició de voltatges de cada banc, que si es necessiten utilitzar diferents VCCO, com a conseqüència, s'hauran d'utilitzar diferents bancs d'E/S.

La següent figura mostra la disposició dels blocs d'E/S (IOB) en 9 bancs d'E/S en una FPGA Virtex 2:



**Figura 12** Distribució dels IOBs en bancs

### 2.2.10 Buffer d'E/S:

El buffer d'E/S d'un IOB és configurable d'acord amb els estàndards d'E/S que es desitja.

La següent taula mostra els diferents estàndards d'E/S disponibles en una FPGA amb les seves respectives tensions d'entrada i sortida.

Estandard	Descripción	Uso	Buffer Entrada	Buffer Salida
LVTTTL	Low-Voltage TTL	Propósito general 3.3V	LVTTTL	Push-pull
LVC MOS	Low-Voltage CMOS	Propósito general 3.3V, 2.5V, 1.8V, 1.5V	CMOS	Push-pull
PCI	Peripheral Component Interconnect	Bus PCI	LVTTTL	Push-pull
GTL	Gunning Transceiver Logic	Bus alta velocidad, backplane	$V_{REF}$	Open Drain
GTL+	GTL Plus	Intel Pentium Pro	$V_{REF}$	Open Drain
HSTL	High Speed Transceiver Logic	Interface con SRAM	$V_{REF}$	Push-pull
SSTL3	Stub Series Terminated Logic 3.3V	SRAM/SDRAM	$V_{REF}$	Push-pull
SSTL2	Stub Series Terminated Logic 2.5V	SRAM/SDRAM	$V_{REF}$	Push-pull
SSTL18	Stub Series Terminated Logic 1.8V	SRAM/SDRAM	$V_{REF}$	Push-pull
<b>Estandares Diferenciales</b>				
LVDS	Low-Voltage Differential Signaling	High speed interface	Diferencial	Diferencial
BLVDS	Bus LVDS	Multipoint LVDS	Diferencial	Diferencial
LVPECL	Low Voltage Positive ECL	High-speed clocks	Diferencial	Diferencial
LDT	Lightning Data Transport	Bidireccional serie/paralelo (Hyper Transport)	Diferencial	Diferencial
Mini-LVDS	Mini-LVDS	Flat Panel Displays	Diferencial	Diferencial
LVDSExt	Extensión de LVDS	Hard Drive interface	Diferencial	Diferencial
RSDS	Reduced Swing Differential Signaling	DVI/HDMI	Diferencial	Diferencial

**Figura 13** Configuració del buffer d'E/S

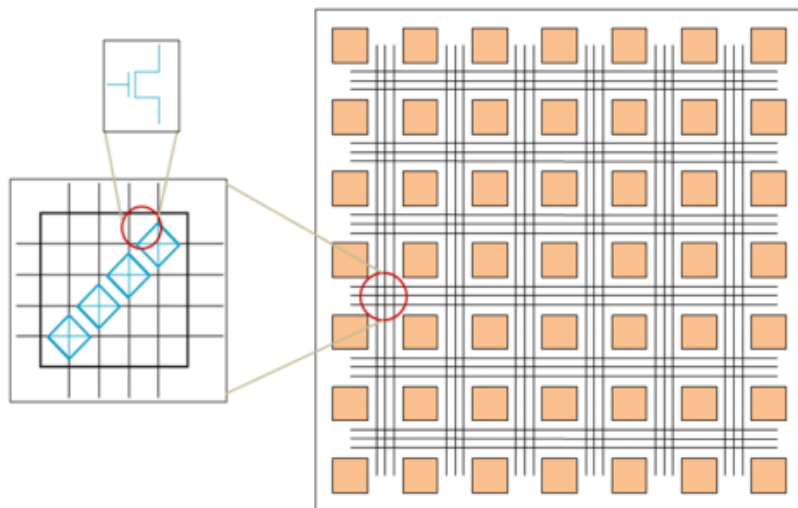
### 2.2.11 Interconnexions a les FPGAs:

Les FPGA, a més de cel·les lògiques programables també inclouen cel·les d'interconnexió programables. Aquestes cel·les defineixen la ruta que seguirà un determinat senyal per dins de la FPGA.

La tecnologia de configuració i l'arquitectura de la cel·la lògica serà la que determinarà la complexitat i l'estructura i d'interconnexió a realitzar.

La connexió interna entre les diferents rutes es realitza mitjançant transistors que actuen com a interruptors per tancar o obrir una ruta.

La següent imatge ens mostra com estan distribuïdes les rutes d'interconnexió entre els diferents CLBs.



**Figura 14** Rutes d'interconnexió entre CLBs

Aquesta capacitat de programabilitat de la interconnexió agrega retards al senyal que ha de passar a través del transistor. Així doncs, si la interconnexió és de gran complexitat i el senyal ha de passar a través de diversos elements d'interconnexió, la suma de tots els retards de cada element pot arribar a ser bastant considerable i s'ha de tenir en compte alhora de dissenyar circuits d'alta freqüència.

Per això existeixen les anomenades rutes llargues, de forma que si un senyal ha de travessar gran part del dispositiu, ho pot fer per aquesta ruta llarga, per així evitar els elements entremitjos que sumarien retard al senyal.



Per la funció contrària existeixen les rutes curtes, que connecten un bloc lògic amb els seus blocs veïns.

Dins de les FPGA també existeixen les anomenades rutes dedicades, que són utilitzades per aquells senyals amb una gran carregabilitat de sortida (fan-out).

Fan-out és definit com el nombre màxim d'entrades de components lògics que poden ésser connectades com la sortida d'un altre component lògic.

Els senyals que utilitzen més habitualment les rutes dedicades són els senyals de rellotge i d'habilitació del rellotge (enable).

Aquestes rutes disposen de buffers especials que estan dedicats a evitar que el senyal es pugui veure afectat i distorsionar-se per causa de la carrega.

Existeix una relació directa entre la quantitat d'interconnexions realitzades i el tamany físic de la FPGA. És per aquest motiu que mai s'implementen físicament totes les interconnexions possibles ni tots els blocs lògics estan interconnectats entre ells.

Per sort per al dissenyador, el software del fabricant del FPGA realitza automàticament totes les interconnexions.

### **2.3 Futur de les FPGA:**

El més probable és que les FPGA acabin essent dispositius amb una utilitat a "mig termini". S'utilitzen principalment per facilitar el disseny i prototip d'ASIC.

A més, s'està ampliant l'àmbit d'aplicacions de les FPGA.

Un exemple de la seva viabilitat seria Intel, que ha invertit 16700 milions de dòlars en la compra d'Altera. Estimacions del mercat indiquen una estimació de 9000-10000 milions de dòlars pel 2020 enfront dels 6000-7000 del 2014 amb un creixement anual del 6-7%.

S'espera que en un futur els costos de les FPGA s'abarateixin i es popularitzin, arribant inclòs a la hibridació entre FPGA i processador, fins arribar a un punt en que el software pugui reconfigurar el hardware en funció de les necessitats.

### **2.4 Fabricants de FPGAs**

El primer circuit FPGA va ser comercialitzat per Xilinx en 1985, encara que la seva difusió és un fet més recent. En els darrers temps, al mercat de les FPGA hi han tres fabricants principals, Xilinx, Altera (que va ser comprada per Intel l'any 2015) i Lattice Semiconductor. També apareixen altres competidors que ofereixen dispositius amb capacitats especials.

### 2.4.1 Lattice Semiconductors

Lattice Semiconductors és una empresa nord-americana especialitzada en la fabricació de dispositius lògics programables d'alt rendiment.

Va ser fundada l'any 1983 i actualment compta amb 700 treballadors i uns ingressos anuals d'uns 300 milions de dòlars.

Lattice és la tercera classificada en la quota de mercat mundial per a les FPGA i per als CPLD i SPLD la segona.

Podem trobar 4 diferents famílies de FGAs dins d'aquesta empresa:

- **ICE40:**

Ice és la marca utilitzada per Lattice Semiconductors per anomenar a la família de FGAs de Baixa potència.

Aquests dispositius es construeixen amb una arquitectura de matriu de blocs lògics programables (PLB) on un PLB és un bloc de 8 cel·les lògiques. Cada cel·la lògica consisteix d'una taula de la veritat de 4 variables (LUT4)

En comparació amb les arquitectures basades en LUT6 (com els de la sèrie Xilinx 7 o els Stratix d'Intel), un dispositiu basat en LUT4 no pot implementar funcions lògiques complexes per el mateix nombre de cel·les. Per exemple, una funció de 7 entrades es podria implementar en 8 LUT4 o 2 LUT6.

Els pins INPUT/OUTPUT estan separats en fins a 4 bancs i suporten múltiples estàndards d'interfaç amb nivells de voltatge d'entre 1.8 i 3.3 V.

- **ECP:**

La família Lattice ECP i Lattice EC

Tenim com exemple la placa d'avaluació Lattice Semiconductors ECP5. Aquesta està dissenyada per permetre als usuaris investigar i experimentar amb les característiques de la matriu de portes programables de camp ECP5-5G.

Aquesta placa consta de 178 INPUTS/OUTPUTS, flash de arrancada integrat, 8 LED per a demostració, 3 pulsadors i 8 interruptors DIP entre altres.

Utilitzen un teixit de silici eficient, on la capacitat d'E/S, la memòria, el traçat de les connexions i la lògica estan optimitzats de forma que s'obtenen les millors característiques a molt bons preus.

- **MACH:**

Les FPGAs Lattice Semiconductors MACH són la pròxima generació de PLD de baixa densitat que inclouen característiques de seguretat millorades i flash d'arrancada dual en el chip.

La família té la major densitat d'E/S en els paquets de menor cost disponibles.

És perfecta per implementar de forma ràpida funcions per al control de sistemes. Són molt útils per a traçadors, estacions base, servidors i emmagatzematge.

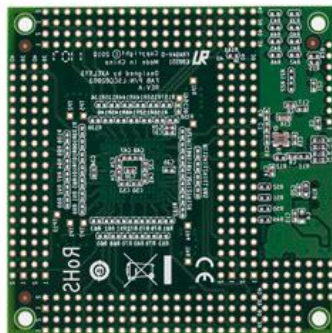
- **CROSSLINK:**

Crosslink és un dispositiu de pont de vídeo programable que admet una gran varietat de protocols i interfases per a sensors i pantalles.

Combina l'extrema flexibilitat d'una FPGA de Baixa potencia, Baix cost i té un espai reduït de fins a 6 mm<sup>2</sup>.

Pot arribar a gastar fins un 50% menys d'energia que la competència.

Té 15 parells d'E/S síncrones de font programable per a la interfaç de càmera i pantalla i més de 5900 LUT amb fins a 81 pins d'E/S disponibles per a múltiples funcions



**Figura 15** FPGA Mach

## 2.4.2 INTEL

Intel és una corporació multinacional nord-americana i una companyia tecnològica amb seu a Santa Clara, California, a Silicon Valley.

És el fabricant de xips de semiconductors més gran i de major valor del món.

També és el inventor de la sèrie x86 de microprocessadors, els processadors que es troben a la majoria d'ordinadors personals.

Va ser fundada el 18 de Juliol de 1968 pels pioners en semiconductors Robert Noyce i Gordon Moore. El nom de la companyia es va concebre com el acrònim entre les paraules integrat i electrònica.

Dins d'Intel, les principals sèries són:

- **STRATIX:**

Les FPGA Stratix combinen una alta densitat amb un alt rendiment que permet habilitar més funcions i maximitzar l'ample de banda del sistema. Això permet als clients portar al mercat productes d'alt rendiment i de Vanguardia de forma més ràpida i per un cost i risc menor.

- **CYCLONE:**

La sèrie Cyclone FPGA està dissenyada per poder satisfer les necessitats de disseny de baix consum i cost.

Cada generació de Cyclone que surt al mercat amb una major integració, major rendiment, menor potencia i un temps de comercialització més ràpid sense augmentar molt els costos.

- **MAX:**

La família de FPGAs MAX revolucionen la integració no volàtil al oferir capacitats de processament avançades en un dispositiu lògic programable de baix cost, amb una encesa del dispositiu instantània i amb un factor de forma petit.

- **ARRIA:**

La família de dispositius Arria ofereix un rendiment òptim amb una eficiència d'energia en uns rangs mitjans.

La família té un ampli conjunt de característiques de bloc de memòria, lògica i processament de senyal digital (DSP) combinades amb la integritat de senyal superior de transceptors de fins a 25.78 Gbps que permeten als dispositius de Arria integrar més funcions i poder maximitzar l'ample de banda del sistema.



**Figura 16** FPGA Stratix 10

### 2.4.3 Xilinx

Xilinx és una companyia nord-americana que està especialitzada en la distribució de dispositius lògics programables. Va ser fundada al Silicon Valley al 1984 i té la seva seu a San José, Califòrnia.

És reconeguda per inventar les FPGA i posseeix la major família de productes FPGAs, incloent les sèries Virtex (amb un elevat rendiment), la sèrie Kintex (rang mig), la sèrie Artix (Baix cost) i la sèrie Spartan (Baix cost) encara que aquesta última ja ha sigut retirada del mercat.

També posseeixen el software computacional Xilinx ISE i Vivado Design Suite.

Les principals sèries de FPGAs de Xilinx són les següents:

- **Sèrie FPGAs Virtex:**

Virtex és la família insígnia de les FPGAs desenvolupades per el fabricant Xilinx.

Alguns membres de la família Virtex com per exemple el Virtex-5QX estan disponibles en envàs que són resistents a la radiació per a que siguin capaces d'operar en ambients hostils com l'espai exterior.

Entre els seus dissenys tenim dissenys que poden variar de tamany entre 28 nm – 20 nm – 16 nm.

- **Sèrie Kintex:**

La família Kintex-7 és la primera família de FPGAs de Xilinx de gama mitjana, que segons la pròpia companyia Xilinx, té el mateix rendiment que la família Virtex 6 a la meitat de preu i amb un consum del 50 % menys d'energia.

- **Sèrie Artix:**

La sèrie Artix-7 ofereix una potència 50% més baixa i un preu del 35% més barat amb comparació amb la sèrie Spartan 6.

Aquesta es basa en l'arquitectura unificada de la sèrie Virtex.

La família Artix està dissenyada per poder atendre les necessitats de factor petit i de rendiment de baix consum d'energia.

- **Sèrie Spartan:**

La sèrie Spartan està dirigida a aplicacions amb una baixa potència, sensibilitat extrema i un volum elevat, com per exemple pantalles, descodificadors, routers inalàmbrics i altres aplicacions.

La sèrie Spartan està basada en una tecnologia de processador de doble òxid, amb 9 capes de metall de 45 nm cada una.

La Spartan 6 va ser comercialitzada al 2009 com una solució de baix cost per a la automoció, comunicacions inalàmbriques, de visualització i de vigilància de vídeo.



**Figura 17** FPGA Virtex 6

#### **2.4.4 Microsemi**

Microsemi corporation és una empresa nord-americana distribuïdora de semiconductors i solucions de sistemes per a sistemes aeroespacials, de defensa, comunicacions, centres de dades i mercats industrials.

Dins dels seus productes podem trobar diferents famílies de FPGA:

- **PolarFire:**

Les FPGAs de Microsemi PolarFire ofereixen fins a un 50% menys de potència requerida que les FPGA SRAM equivalents. Ofereixen baixa potència a densitats de rang mitja gràcies als transceptors Serializer/Deserializer de 12.7 Gbps per a la seva seguretat i confiabilitat.

La família és ideal per a una ampla gama d'aplicacions dins de xarxes d'accés amb filferro i infraestructura cel·lular, mercats de defensa i aviació comercial i la indústria 4.0 que inclou mercats d'automatització industrial.

- **IGLOO2:**

Les FPGA IGLOO2 ofereixen LE de 5K-150K amb un subsistema de memòria d'alt rendiment, amb fins a 512 KB de memòria flash incorporada, 2x32 KB de memòria SRAM, dos motors d'accés directe a memòria (DMA).

També ofereixen una reducció de potencia que pot variar entre el 20-40% amb un nivell de seguretat comprovat i protecció contra sobre construccions i clonacions..

- **PROASIC3:**

La sèrie FPGA ProAsic, que inclou ProAsic3, ProASIC3 nano i ProASIC3L ofereix un avanç en rendiment densitat i característiques per a aplicacions amb un elevat volum, cosa que les fa molt exigents.

Els dispositius ProASIC3 són compatibles amb el nucli IP de processador suau ARM-CORTEX-M1, que ofereix els beneficis de la programabilitat.

La família ProASIC3 està basada en tecnologia flash no volàtil i admet de 100 a 35k d'elements lògics i fins a 620 INPUTS/OUTPUT.

- **IGLOO:**

La família de FPGA IGLOO està dissenyada per satisfer la demanda de Baixa potencia i requisits de “petjada” petita de la electrònica portàtil actual i conscient de l'energia.

La família de Baix consum IGLOO admet elements lògics fins a 35 k amb fins a 504 Kbits de SRAM de doble port, 6 PLL integrats i fins a 620 I/O d'usuari.

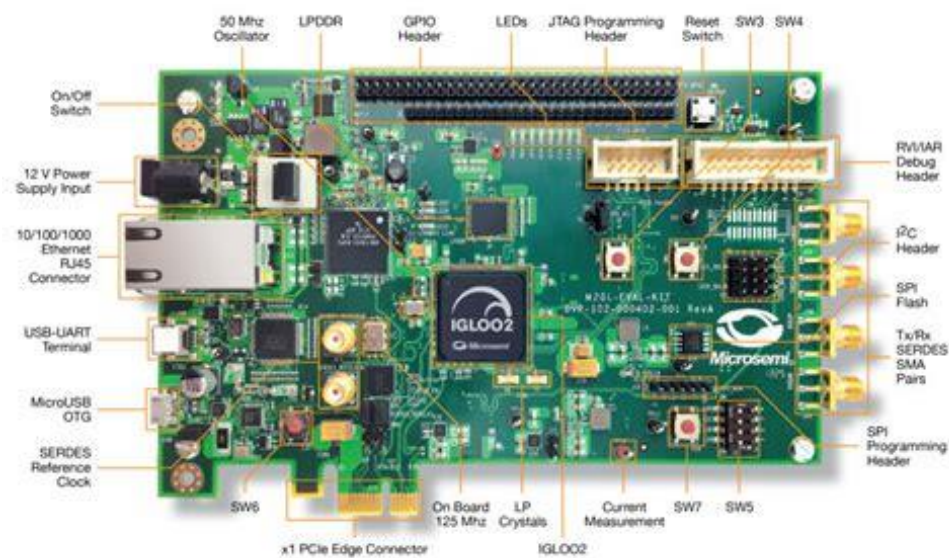
Els dispositius de la família IGLOO estan disponibles en grandària reduïts (3x3,4x4,5x5,6x6 i 8x8 mm), alta densitat, escala de viruta i paquets quàdruples sense plom.

- **FUSION:**

Les FPGA de senyal mixta Fusion integren grans blocs de memòria flash configurables, circuits integrats de generació i gestió del rellotge i lògica programable basada en flash d'alt rendiment en un dispositiu monolític.

La innovadora arquitectura Fusion FPGA es pot utilitzar amb el processador ARM-CORTEX-M1 de 32 bits amb un rendiment maximitzat.

Els dispositius Fusion són la primera plataforma FPGA de senyal mixta al mon.



**Figura 18** FPGA Igloo2

## 2.5 Llenguatge de Programació

Les FPGA no es “programen” en el sentit estricte, sinó que utilitzen HDL (Hardware Description Language). HDL és un llenguatge especialitzat que s'utilitza per a definir l'estructura, disseny i operació de circuits electrònics i electrònics digitals. Per tant, aquests llenguatges fan possible una descripció formal d'un circuit electrònic i possibiliten la seva anàlisi automàtica i la seva simulació. Un inconvenient d'aquests llenguatges HDL és que són específics per a cada FPGA i estan lligats a l'arquitectura de cada fabricant. No obstant això, com a avantatge trobem que tenen una corba d'aprenentatge elevat pel fet que té un grau d'abstracció molt baix.



Existeixen diversos llenguatges de programació per a poder dissenyar circuits digitals per a les FPGAs. Alguns d'aquests llenguatges són: VHDL, Verilog, System Verilog, SystemC, Handel-C, Pure C/C++, Simulink o LabVIEW.

Verilog i VHDL són dels primers llenguatges i els més utilitzats per a la síntesi i simulació de sistemes digitals per a la FPGA. A mesura que les FPGA van anar creixent en potència i velocitat, van anar sorgint diverses versions de llenguatges d'un alt nivell coneguts com C i C++ per a les FPGA. Altres opcions per un nivell més elevat d'abstracció serien Simulink i LabVIEW.

En aquest treball explicarem breument dos d'aquests llenguatges: VHDL i Verilog.

### **2.5.1 VHDL**

VHDL és l'acrònim que representa la combinació de VHSIC (Very High Speed Integrated Circuit) i HDL (Hardware Description Language).

Originalment, VHDL va ser patrocinat pel departament de defensa de E.E.U.U i posteriorment va ser transferit al IEEE (Institute of Electrical and Electronics Engineers).

En un principi, VHDL va sorgir com un llenguatge de simulació de circuits digitals i per al disseny s'utilitzaven altres eines com Netlist i esquemàtics.

A mesura que els circuits digitals es van anar fent més complexos, va sorgir la necessitat de poder descriure el circuits amb un alt nivell d'abstracció. Va ser llavors quan es va començar a utilitzar el VHDL per al disseny de circuits digitals.

El llenguatge VHDL és utilitzat per a la programació de dispositius de lògica programable com les FPGA i els CPLD.

Encara que VHDL és un llenguatge de programació com qualsevol altre respecte a les estructures bàsiques de programació, també és important mencionar que la execució dels codis en VHDL no és la mateixa a la execució de codis en microprocessadors o microcontroladors.

Quan s'està desenvolupant un programa per a un microcontrolador, el programador pot tenir la seguretat de que les variables utilitzades a cada línia de codi estan inicialitzades o han estat utilitzades, inclòs pot utilitzar una mateixa variable en dos línies de codi diferents i tenir la seguretat de que aquestes no entren en conflicte, ja que les línies de codi no s'executen de manera simultània.

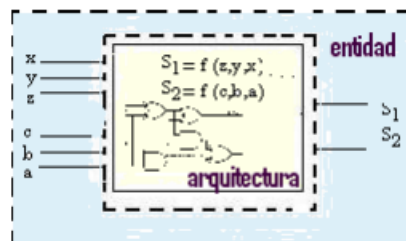
En el cas de la programació VHDL això no està tan clar, ja que la forma en la que s'executa el codi és en funció dels canvis a les variables i senyals que conformen el sistema.

VHDL significa llenguatge de descripció del hardware. Això indica que el que realment s'està programant són les connexions físiques que s'estableixen dins del dispositiu de lògica programable i que per tant, si volem guardar informació en una variable, ens hem d'assegurar que no la utilitzem a cap altre lloc, ja que estariem provocant un curtcircuit.

El llenguatge VHDL és un estàndard de domini públic anomenat IEEE 1076-1993. Per tant, no depèn de cap fabricant o dispositiu, és independent i això també fa que es puguin reutilitzar els dissenys. Com avantatge també trobem que és un disseny jeràrquic, per la qual cosa es manté un ordre i es segueixen unes certes regles jeràrquiques.

El llenguatge VHDL està compost per dos parts fonamentals: l'entitat i l'arquitectura:

- Entitat VHDL: Declaració dels vectors del mòdul, generalment entrades i sortides.
- Arquitectura VHDL: Descripció detallada de l'estructura interna o el comportament del mòdul.



**Figura 19:** Arquitectura VHDL

Com podem veure al diagrama de blocs, l'entitat implica els llaços que té un sistema per a ser excitat i entregar la informació a altres, És a dir, implica tot l'extern al diagrama, mentre que l'arquitectura descriu tot el que hi ha a l'interior del bloc.

En el desenvolupament jeràrquic es defineixen entitats de nivell baix o de nivell alt. Aquestes es defineixen depenent de si l'entitat necessita altres entitats per poder definir el seu comportament o no.

Per a realitzar un programa en VHDL, s'han de definir tant l'entitat com l'arquitectura. Primer es declara l'entitat i l'arquitectura la segueix.

La sintaxis general per declarar tant entitat com arquitectura en VHDL és de la següent forma:

- Entitat:

La declaració bàsica té els següent propòsits: Nombrar a l'entitat i la declaració dels ports, que defineix els senyals d'interfaç externa que es manifestaran.

Es representa de la següent forma:

```
ENTITY nom de l'entitat  
  
IS PORT (nom dels senyals : mode tipus-  
senyal ;  
  
nombre-senyals : mode tipo-senyal;  
  
.....  
  
nombre-senyals : mode tipo-senyal) ;  
  
END nombre-identitat ;
```

### 2.5.2 Verilog

Verilog és un llenguatge de descripció de hardware utilitzat per poder modelar sistemes electrònics. Aquest llenguatge suporta el disseny, prova i implementació de circuits analògics, digitals i senyal mixta a diferents nivells d'abstracció.

Els dissenyadors de Verilog volien un llenguatge amb una sintaxis similar a la del llenguatge C per a que els hi resultés familiar als enginyers i així fos acceptat d'una forma més ràpida.

A diferència del llenguatge C, Verilog utilitza Begin/End en lloc de Claus per definir un bloc de codi.

D'altra banda, la execució de sentències en Verilog no és estrictament lineal.

Un disseny Verilog consisteix en una jerarquia de mòduls. Aquest mòduls estan definits com conjunts de portes d'entrada/sortida i bidireccionals. Internament un mòdul conté una llista de cables i registres. Les sentències concurrents i seqüencials defineixen el comportament del mòdul, descrivint les relacions entre els ports, cables i registres.

- **Programació en Verilog:**

Les característiques principals dels mòduls de Verilog són les següents:

- Cada mòdul té una sèrie d'entrades i sortides amb les quals es poden interconnectar altres mòduls, encara que aquests no tinguin ni entrades ni sortides, com és el cas dels testbenches.
- En Verilog no existeixen les variables globals.
- Fora dels mòduls només hi ha directives de compilador.

Verilog utilitza un sistema digital que està compost per la interconnexió de diversos mòduls. L'estructura general d'aquests mòduls és la següent:

```
module <nom> (<senyals>;  
  
    <declaració de senyals>  
  
    <funcionalitat del modul>  
  
endmodule
```

Un exemple d'un mòdul que suma un array de 3 bits indicant si hi ha carry o no seria el següent:

```
module sum(A, B, R, C);  
    input  [2:0] A, B;  
    output [2:0] R;  
    output C;  
  
    reg [2:0] R;  
    reg C;  
  
    always @(A or B)  
    begin  
        {C,R} = A + B;  
    end  
endmodule
```

Els arguments del mòdul poden ser de tres formes diferents:

- Input
- Output
- Inout (bidireccional)

### **Tipus de dades:**

En Verilog s'utilitzen principalment dos tipus de dades:

- **Nets:** Utilitzades per representar connexions estructurals entre components. No permeten emmagatzemar dades dins seu. El més utilitzat d'ells és el tipus wire
- **Registers:** Representen variables que sí que tenen la capacitat d'emmagatzemar dades. Mantenen el valor d'una dada fins que no se li canvia el valor de nou. Els tipus de registres més comuns són el reg i integer. Aquests últims s'utilitzen per la creació de testbenches.

### **Processos:**

El concepte de processos que s'executen en paral·lel és una de les característiques bàsiques de Verilog, essent un dels principals aspectes que el diferencien respecte als llenguatges de programació comuns com el C.

Tota descripció de comportament de llenguatge Verilog s'ha de declarar dins d'un procés, encara que existeixen algunes excepcions.

En Verilog existeixen dos tipus de processos:

- **Initial:** Aquest és un procés que s'executa una sola vegada al temps 0. Aquest procés està dedicat a la realització de testbenches.
- **Always:** Aquest tipus de procés s'executa contínuament a mode de laç. Tal i com indica el seu nom, aquest s'executa sempre. La seva execució està controlada per un temporitzador (asíncrona) o per events. En aquest últim cas, si el bloc s'activa per una llista d'events se li denomina "llista sensitiva". Si volem activar-lo mitjançant un temporitzador, la senyal de rellotge denominada clock s'ha d'incloure dins de la llista sensitiva, mitjançant el caràcter @ tal i com es veu a continuació:

```
always @ (posedge clk)
```

## **3. HARDWARE LLIURE /SOFTWARE LLIURE**

### **3.1 Concepte**

Per la gran majoria de gent, el terme anglès hardware és estrany, res més que un terme general per descriure dispositius tecnològics. En informàtica, es denomina hardware al conjunt d'elements materials que componen el computador.

El concepte de Hardware lliure, tal com el software lliure es refereix a la possibilitat d'utilitzar dispositius lliurement, és a dir, que hi ha llibertat d'ús, d'estudi i modificació i distribució. Això no vol dir que hagi de ser gratuït, sinó que qualsevol pot accedir a la seva documentació i eines utilitzades per al seu disseny.

### **3.2 Avantatges i inconvenients d'eines SF/HW lliures**

Les avantatges de l'ús del software/hardware lliure són:

- Independència del fabricant: Els desenvolupadors de hardware poden desenvolupar els seus sistemes amb autonomia.
- Accés al coneixement: Es fomenta la creativitat, l'interès i l'esperit científic.
- Noves Aplicacions: S'obra la possibilitat a nous usos de la FPGA que no han estat previstos pel propi fabricant.
- Repositoris de Hardware lliure: Els dissenys de hardWare lliure poden ésser compartits mitjançant repositoris.
- Cost: Es redueixen els costos ja que les tecnologies lliures són més barates que les privatives.
- Participació de la comunitat: tots podem participar fent créixer i millorant les eines.

Quant a les limitacions podem citar les següents:

- Els dissenys són únics, el que dificulta la seva reproducció.
- La disponibilitat dels chips és sovint difícil.
- La fabricació de hardware porta implícit la infraestructura de disseny, simulació i implementació, cosa que no succeeix amb el software lliure.

### **3.3. Antecedents. El món Maker**

Als anys 1990 sorgeix Open Desing Circuits, una pàgina web creada per Reinoud Lamberts. Open Desing Circuits va ser el primer en proposar la creació d'una comunitat de disseny de hardware amb un esperit de software lliure. En teoria aquesta pàgina permetria l'intercanvi de dissenys lliures de la mateixa forma que els programes poden

ser intercanviats. Però a la pràctica, mai es va agregar cap disseny lliure com a conseqüència de la carència de software lliure (que en aquell moment no existia).

Aquesta va ser la primera vegada que un gran grup de gent va discutir seriosament sobre les avantatges i desavantatges respecte el hardware lliure.

Des de 2005 la influència del moviment maker, una ampla comunitat d'enginyers, científics, estudiants i aficionats que propugnen l'ús de tecnologies lliures i obertes, tant a nivell de software com de hardware, ha donat lloc a la creació de plataformes low-cost com Arduino, Raspberry Pi o les impressores 3D.

Encara que les PPGA es coneixen des de fa 30 anys i eren eines molt útils, molta gent no les utilitzava, perquè tradicionalment els fabricants es van basar en el codi privat i mai han alliberat ni el programari ni les especificacions dels formats dels seus bitstreams, basant-se en codi privat i hardware tancat.

No va ser fins l'any 2015, quan l'enginyera austríaca Claire Wolf (abans coneguda com Clifford Wolf), va fer enginyeria inversa de les FPGA Ice40 de Lattice i va desenvolupar una sèrie d'eines de software lliures amb la capacitat de programar els models de FPGA Ice40 LP/HX 1K/4K/8K del fabricant Lattice Semiconductor.

També s'admeten les peces UltraPlus ICE40, incloent els oscil·ladors, SRAM, RGB i DSPs.

Claire Wolf va crear el projecte IceStorm on va obtenir la primera toolchain de codi lliure que va permetre programar una FPGA.

### **3.4. El projecte ICESTORM**

Aquest conjunt d'eines de software va acabar formant el que es coneix com la toolchain del projecte ICESTORM, on es va alliberar la primera aplicació que permet passar de Verilog al bitstream, utilitzant únicament eines lliures.

Aquesta permet mitjançant una sèrie de seqüències preestablertes d'ordres i programes sintetitzar, implementar i guardar els fitxers bitstream en la FPGA.

La síntesi es realitza amb el programa Yosis, que converteix el codi de descripció de hardware (en llenguatge Verilog) a una "netlist" en un format BLIF (Berkeley Logic Interchange Format).

La implementació es fa amb el programa nextpnr, que accepta com a entrades fitxers que contenen blocs de 0 i 1 per als bits de configuració de cada cel·la de la FPGA.

Finalment, el programa icepack transforma el fitxer a un format ASCII generat pel programa nextpnr en un fitxer en format binari (.bin) i el programa iceprog permet que aquesta informació en binari pugui ser gravada dins de la FPGA.

El fet de disposar d'una toolchain lliure que permet gravar informació dins d'una FPGA de la família Ice40, coincidint amb l'augment del moviment maker, ha provocat l'aparició de molts projectes de desenvolupament de hardware lliure al voltant de les FPGAs d'aquesta família, on es disposa de tota la informació per poder-la replicar (arxius gerber, BOM, esquemes, etc.).

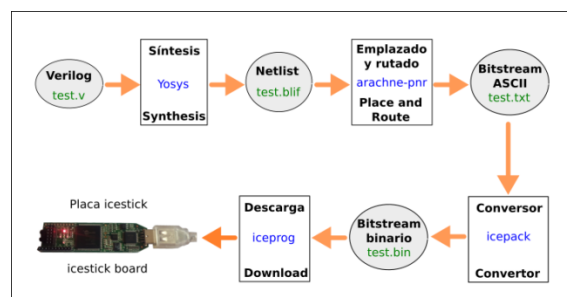
En aquest context és on han aparegut diverses plaques d'entrenament, com són la Icezum Alhambra II, la tiny FPGA, iCEBreaker, icoBoard, entre d'altres.

Un altre factor que ha provocat l'augment de la popularitat de les FPGA de la família Ice40 ha estat la possibilitat d'utilitzar la toolchain del projecte IceStorm mitjançant l'editor visual Icestudio.

IceStudio és un projecte de software lliure que va començar la empresa BQ l'any 2016 i que ha estat continuat per la comunitat maker #FPGAWars, amb Juan González-Gómez (Obijuan) al capdavant.

Aquest editor permet dissenyar sistemes i blocs digitals complexos a partir de portes lògiques i dispositius combinacionals/seqüencials i es pot anar incrementant la complexitat dels blocs aplicant bottom-up fins a obtenir un disseny del sistema digital complet. També permet incloure amb facilitat blocs de codi Verilog per poder simular el comportament del bloc.

Un cop el disseny del circuit digital ha estat finalitzat, IceStudio internament utilitza una toolbox de software lliure anomenada Apio per verificar, sintetitzar i carregar el disseny desenvolupat en la FPGA mitjançant l'aplicació de forma seqüencial dels programes de la toolchain del projecte IceStorm.



**Figura 20** Toolchain del Projecte IceStorm



### 3.5 Plaques de la família Ice40:

La línia iCE40 de Lattice són les primeres FPGAs del mercat per a les quals existeix un flux de desenvolupament completament basat en software lliure.

Originalment, les iCE40 van ser creades per l'empresa Silicon Blue Technologies i comprada més tard per Lattice. El seu objectiu principal va ser proveir solucions petites de baix cost i consum. Aquestes FPGAs posseeixen una estructura molt simple en comparació amb les de Xilinx i Altera.

Dins de la família Ice40 trobem tres tipus de models:

- **Ice40 UltraPlus:** Amb la creixent complexitat dels sistemes utilitzats per alimentar llars, fàbriques i ciutats intel·ligents, l'iCE40 UltraPlus FPGA pot resoldre problemes de connectivitat amb una àmplia varietat d'interfases i protocols i proporcionar recursos computacionals de baixa potència amb nivells elevats d'intel·ligència.

Amb taules de cerca (LUT) de 5k pot implementar xarxes neuronals per analitzar la coincidència de patrons necessària per portar la intel·ligència al límit.

Característiques:

- Arquitectura lògica flexible amb 2800 o 5280.
- 4 LUT d'entrada
- I/O personalitzables
- Entre 80 kbits i 1 Mbit de memòria integrada
- Corrent activa de 1-10 mA per la majoria d'aplicacions.
- Processament de la senyal d'alt rendiment, utilitza blocs DSP amb funcions de multiplicar i acumular
- .



**Figura 21** FPGA Ice40 UltraPlus

- **iCE40 Ultra/UltraLite:** a diferència de les FPGA tradicionals, la majoria de dissenys de la família Ice40 s'executen amb un nivell de potencia mW d'un sol dígit.

Característiques:

- Arquitectura lògica flexible amb fins 3520 LUT de 4 entrades.
- Fins a 26 I/O per a interfaços personalitzables.
- Fins a 80 kBits de memòria distribuïda integrada.
- Corrent de repòs, tan baixa com 35uA i corrent activa d'entre 1-10 mA per la majoria d'aplicacions.
- Blocs de SPI i I2C endurits per interactuar en una gran varietat de sensors i perifèrics.

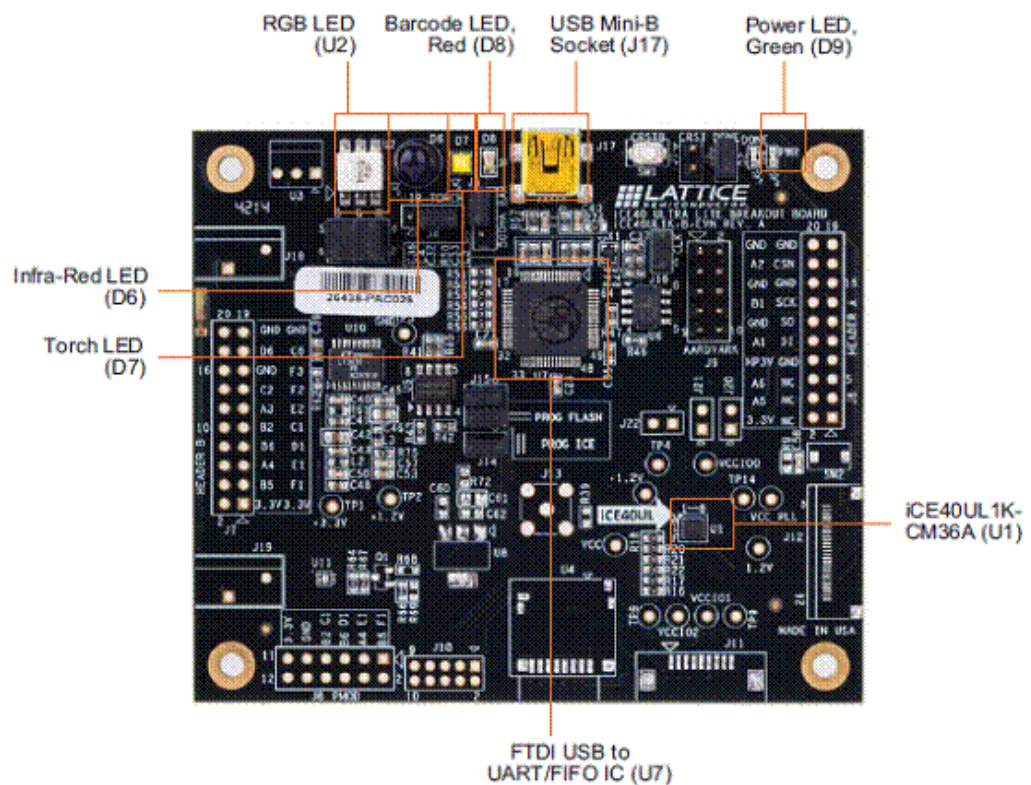


Figura 22 FPGA Ice40 Ultra

- **Ice40 LP/HX/LM:** Dissenyats des de zero per complir una potència baixa a partir de 25 uW, aquests dispositius de la família Ice40 maximitzen la vida útil de la bateria i minimitza el consum d'energia.

Amb unes mesures de només 1.40 mm X 1.48 mm X 0.45 mm, els dispositius Ice40 LP/HX/LM es poden col·locar en els mòduls amb espai més limitat.

Característiques:

- Disponible en tres sèries amb LUT que va des de 384 a 7680. Baixa potència (LP), baixa potència amb IP incorporada (LM) i alt rendiment (HX).
- Nuclis de I2C i SPI integrats que permeten la configuració flexible del dispositiu a través de SPI.
- Fins a 128 Kbits sysMEM™ Embedded Block RAM.
- S'adapta a les aplicacions amb espai limitat.



**Figura 23** FPGA Ice40 LP/HX/LM

## 4. PLACA ICEZUM ALHAMBRA II

La placa Icerzum Alhambra II és una placa electrònica Open FPGA compatible amb Arduino/zoom. Basada en el paradigma de Hardware lliure, la Icezum Alhambra va ser dissenyada utilitzant eines de codi obert i tecnologies lliures. (FreeCad, KiCad, InkScape i LibreOffice) i és compatible amb la toolchain del projecte IcesTorm de Clifford Wolf.

El desenvolupament de l'Icezum Alhambra s'inicia l'any 2015 en el departament de robòtica i innovació de BQ de la mà del dissenyador Eladio Delgado, un enginyer que viu en un petit poble de Granada, d'aquí el nom de la placa, Alhambra. Es pretenia obtenir una placa FPGA, totalment oberta, similar a les plaques Arduino/ZOOM, amb la idea que pogués ser utilitzada en entorns educatius i es poguessin crear dissenys de circuits digitals o experiments de robòtica, utilitzant una FPGA en comptes d'un processador.

A principis de 2016 es va crear el repositori de bqlabs de la placa icezum en github publicant-se tots els arxius de la V1.0 i es construeixen els primers prototips, que van ser muntats manualment.

Més endavant, el finançament de BQ es deté i el projecte és continuat amb el suport de la comunitat maker #FPGAwards, liderada per Juan González Gómez, més conegut com Obijuan. Es crea l'organització github de FPGAwards i tots els repositoris relacionats amb openfpgas es passen allà.

La versió 1.0 es va fer en Altium i posteriorment Obijuan la va migrar a Kicad, una eina lliure de disseny de PCBs, impulsada pel CERN (Organització Europea de Recerques Nuclears). La versió real és l'Icezum Alhambra II. Tots els seus arxius d'arxiu font i fabricació estan disponibles en el repositori, on es poden obtenir totes les dades del disseny, esquemes, pcb, pinout, BOM, models 3D dissenyats amb Freecad.

Fabricada per l'empresa espanyola AlhambraBits, el seu cost (60€), la facilitat per a la seva adquisició, l'accés total als seus plans i les característiques tècniques van determinar l'opció d'aquesta placa per al projecte de la plataforma de desenvolupament de baix cost de la UdL.

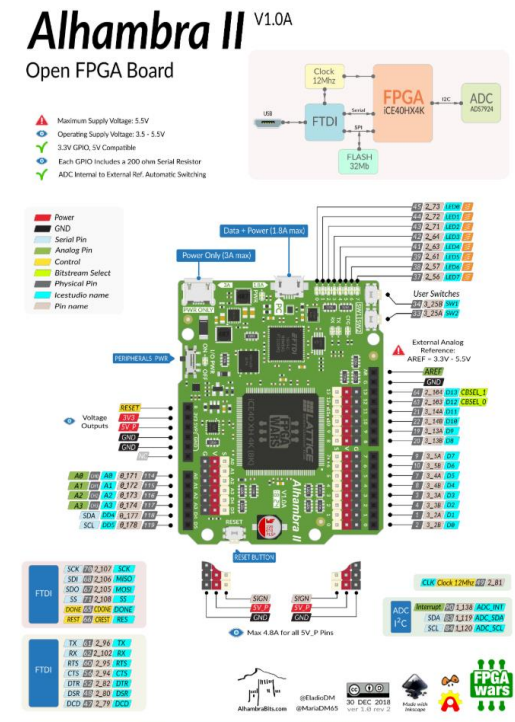


Figura 24 FPGA Alhambra II Pinout

## 4.1 Característiques

La placa Icezum Alhambra II conté una FPGA ICE40HX4K de Lattice Semiconductor .

Les característiques tècniques principals són les següents:

- Placa de desenvolupament FPGA iCE40HX4K-TQ144 de Lattice Semiconductor.
  - Compatible amb la cadena d'eines de font oberta [icestorm] i Icestudio.
  - Arduino com a placa: pinout similar que Arduino.
  - Pot reutilitzar la majoria dels escuts disponibles.
  - Controla els teus robots / printbots des d'una FPGA.
  - Oscil·lador MEMS de 12 MHz.
  - Interruptor ON/\*OFF (apagui el robot mòbil fàcilment).
  - 8 Leds SMD de propòsit general (LED0...LED7).
  - 2 polsadors tipus push-\*button de propòsit general (SW1 i SW2).
  - 32 Mb de Memòria Flaix per poder guardar fins a 30 bitstreams o dades de l'usuari.
  - 20 pins d'entrada/sortida (E/S) (S0...D13, A0...A3, DD4 i DD5) a 3.3v (5v tolerant).
- Com a entrada accepta nivells entre 3.3V i 5V, com a sortida genera 3.3V.
- Resistència de 200Ω en sèrie en tots els pins de E/S per activació directa de Leds.
  - Convertidor 12 bits A/D (4 canals).

- Els pins de selecció de bitstreams per a cold boot (a triar entre 4 bitstreams que prèviament hagin estat gravats en la memòria Flash) estan accessibles mitjançant els pins de E/S de propòsit general.
- Reguladors commutats de 1A per a les alimentacions d'1.2 V i 3.3 V, la qual cosa permet activar els PLLs i treballar a velocitats molt majors.
- Oscil·lador MEMS de 13 MHz.
- Botó de reset tipus push-button.
- Font d'alimentació USB. Dos connectors. Fins a 4.8A d'entrada.
- Pins analògics (encara que el convertidor A/D incrustat i2c).
- El dispositiu USB FTDI 2232H permet la programació FPGA i la interfície UART a un PC.

Si la FPGA iCE40HX4K es programa utilitzant la toolchain lliure del projecte IceStorm, fa possible que el número de cel·les bàsiques lògiques de què es disposin sigui de 7680, és a dir, el doble que si es programés mitjançant el software privatiu del fabricant Lattice. D'aquesta forma, utilitzant les eines lliures es poden sintetitzar sistemes digitals molt més complexos.

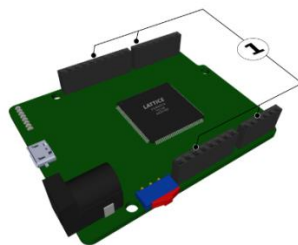
## 4.2 Connectors

Aquests connectors són:

1. Connectors de E/S femella de 5 Volts
2. Connectors de E/S mascle de 5 Volts, amb alimentació i massa

## 4.3 Connectors femella de 5V

Són els típics connectors de Arduino que serveixen per a connectar cables mascle i escuts d'expansió. Es troben en la part superior i inferior de la placa, com es mostra en la figura:



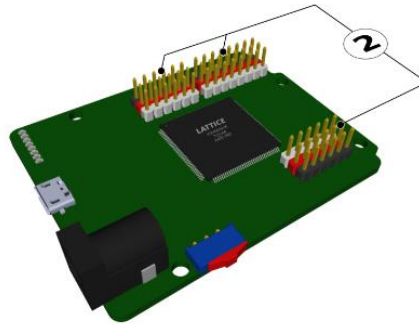
**Figura 25** FPGA Alhambra II Pins Femella

En total hi ha 31 pins, dels 20 són de E/S de 5V. Els pins de E/S superiors es denominen D13 - D0 en Icestudio, i els inferiors, DD5 - DD0. Tots aquests pins tenen en comú que són de 5V. Com els pins de la FPGA són de 3.3 volts, es passen a través d'un convertidor

bidireccional de 3V3 - 5v\*. Per això, no hi són connectats directament als pins de la FPGA, i les seves característiques les determinen aquests convertidors.

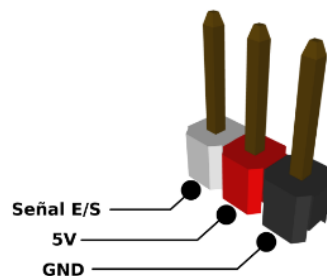
#### 4. 4 Connectors d' E/S mascle de 5V amb alimentació i massa

Són 20 pins de E/S, mascles, duplicats dels 20 pins E/S dels connectors femella de 5v



**Figura 26** FPGA Alhambra II Pins Mascle

En cadascun d'aquests pins, un inclou addicional d'alimentació (+5v) i massa (GND). En total són tres pins però no s'especifica per colors. El pin blanc amb el senyal de E/S, el vermell l'alimentació de 5v i el negre la massa (GND).



**Figura 27** FPGA Alhambra II Pins Mascle

Tenen aquesta disposició per a poder connectar directament servos i sensors. Es denominen igual que les seves duplicats femella: D0 - D13 per als de la part superior i DD0 - DD5 en la part inferior. En aquesta foto es mostra un servo connectat al pin D0 i un sensor d'infrarojos

## 4. 5 Bill of Material

A continuació es mostra la taula 1 on està recollida en una taula la llista dels materials i components necessaris per la seva fabricació.

Dins de la taula trobem la següent informació d'esquerra a dreta: Nom del component, Quantitat, Tipus de component( Resistència, Condensadors, etc), Valor, Footprint ,Fabricant

C1, C2, C4, C5, C7, C8, C9, C10, C11, C12, C13, C14, C15, C16, C17, C18, C20, C21, C22, C23, C25, C26, C27, C28, C29, C30, C32, C33, C34, C35, C36, C38, C39, C40, C46, C48, C49	37	Capacitor SMD	100nF ±10% 16V	C0402	Kemet
C3, C6, C19, C24, C31, C37, C43	7	Capacitor SMD	10uF ±10% 10V	C0603	Taiyo Yuden
C41, C45, C47	3	Capacitor SMD	10uF ±10% 10V	C0805_1.25_HD	WE
C42, C44	2	Capacitor SMD	100pF ±5% 25V	C0402	WE
C50	1	Capacitor SMD	220uF 10V	C_POL_6.3x5.8_WE	WE
D1, D5, D6	3	Schottky Diode	200mA 30V	SOD523	On Semiconductor
D2	1	Protection, USB ESD		SOT563	Würth
D3	1	Protection, Transient Suppression Diode		DO-214AC-SMA_Unidir	WE
D4	1	Zener Diode	3.6V	DO-219AC	Vishay
FB1, FB2, FB3, FB4, FB5	5	Ferrite Bead	330R 0,080R 1700 mA	L0603	Murata
J1, J5	2	Connector, 8 pin Socket Header 2.54mm	8 PIH	1x8 Socket Header THT 2.54mm PIH	Samtec
J2, J6	2	Connector, 6 pin Socket Header 2.54mm	6 PIH	1x6 Socket Header THT 2.54mm PIH	Samtec
J3, J7	2	Connector, 1x6 pin Header 2.54mm		1x6 Pin Header 2.54 WHITE PIH	Samtec
J4	1	Connector, 1x8 pin Header 2.54mm		1x6 Pin Header 2.54 WHITE PIH	Samtec
J8, J12	2	Connector, 1x6 pin Header 2.54mm		1x6 Pin Header 2.54 WHITE PIH	Samtec
J9, J13	2	Connector, 1x6 pin Header 2.54mm		1x6 Pin Header 2.54 RED PIH	Samtec
J10	1	Connector, 1x8 pin Header 2.54mm		1x8 Pin Header 2.54 RED PIH	Samtec
J11	1	Connector, 1x8 pin Header 2.54mm		1x8 Pin Header 2.54 RED PIH	Samtec
J14	1	Micro USB 2.0 SMT Type B Horizontal 5 Contacts WR-COM		629105136821	WE
J15	1	Micro USB 2.0 SMT Type B Horizontal 5 Contacts High Current WR-COM	4,7uH 1,7A	629105150521	WE
LD0, LD1, LD2, LD3, LD4, LD5, LD6, LD7, LD9	9	LED	Green	LED0603	Diodes Incorporated
LD8, LD10	2	LED	Yellow	LED0603	Diodes Incorporated
LD11, LD12	2	LED	Blue	LED0603	Diodes Incorporated
Q1	1	P-Channel MOSFET		U-DFN2020-6	
R1, R2, R19, R20, R21, R28, R29, R31, R32, R36, R37, R38, R39, R41, R42, R43	16	Resistor, SMD	10k ±1% 0.063W	R0402	Vishay
R3, R23, R33, R47, R50	5	Resistor, SMD	1k ±5% 0.063W	R0402	Vishay
R4, R5, R6, R7, R8, R9, R10, R11, R12, R13, R30	11	Resistor, Array 4x1	100R 5% 0,0625W, 100R ±5% 0,0625W	R0402	Vishay
R14	1	Resistor, SMD	10K ±5% 0,0625W	R0402	Yageo
R15, R34, R35	3	Resistor, SMD	470R ±5% 0,0625W, 470R 5% 0,0625W	R0402	Yageo
R16, R17, R25, R26, R27, R46	6	Resistor, SMD	100R ±5% 0.063W	R0402	Yageo
R18, R22, R44, R45	5	Resistor, SMD	2k2 ±5% 0.063W	R0402	Yageo
R24, R48, R49	3	Resistor, SMD	2k2 ±5% 0.063W	R0402	Vishay
R40	1	Resistor, SMD	2k2 ±5% 0.063W	R0402	Vishay
SW1, SW2, SW3	3	Micro miniature pushbutton surface Switch			WE
SW4	1	WS-SLSU 6.7x2.7 mm SMD Right Angled, boss design			WE
U1	1	iCE40 HX4K Series FPGA Family, 3520 Logic Cells, 1.2 V Core, 80 K Embedded RAM Bits, 144-pin TQFP, Pb-Free, Tray		TQFP144_HX4K_FPGAware	Lattice
U2	1	NOR Flash Memory, serial	32Mb	DFN-8 6x5mm	Microchip
U3	1	Dual High Speed USB to Multipurpose UART/FIFO IC		FTDI_QFN-64	Microchip
U4	1	EEPROM, 2K Microwire Compatible Serial, 8 or 16 bit word size, 2.5-5.5 Vcc range	2Kb	MSOP8	ON Semiconductor
U5	1	Oscillator, Low-Power CMOS, 4 pin, 1.8-3.3V	12MHz	OSC 3.2x2.5	Texas Instruments
U6	1	LDO Regulator, 5pin, SOT-23		SOT23-5_M	
U7	1	ADC Converter, 12-Bit, 4-Ch, I2C		QFN16	
U8	1	Dual Step-Down DC/DC converter 1A, 1.5 MHz		WDFN_12_3x3	Diodes
U9	1	Voltage supervisor, Reset active high, 200ms	2.93V	SOT23-3	Diodes Incorporated

**Taula 1** Bill of Material Icezum Alhambra II



## 4.6 Schematics

Les següents figures mostren els schematics necessaris per a poder fabricar la Iczum Alhambra II amb les seves corresponents connexions.

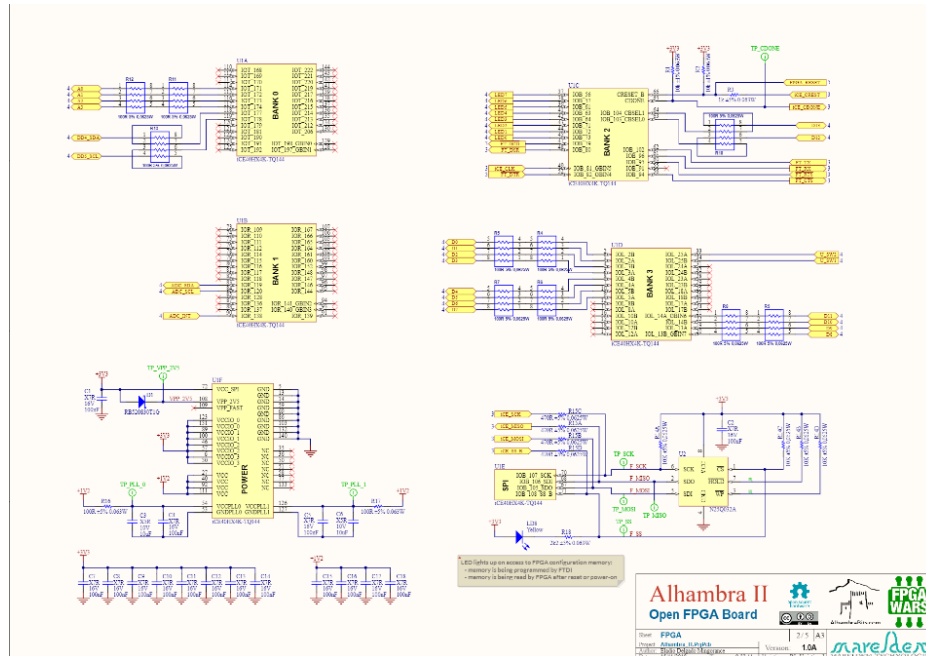


Figura 28 Schematics

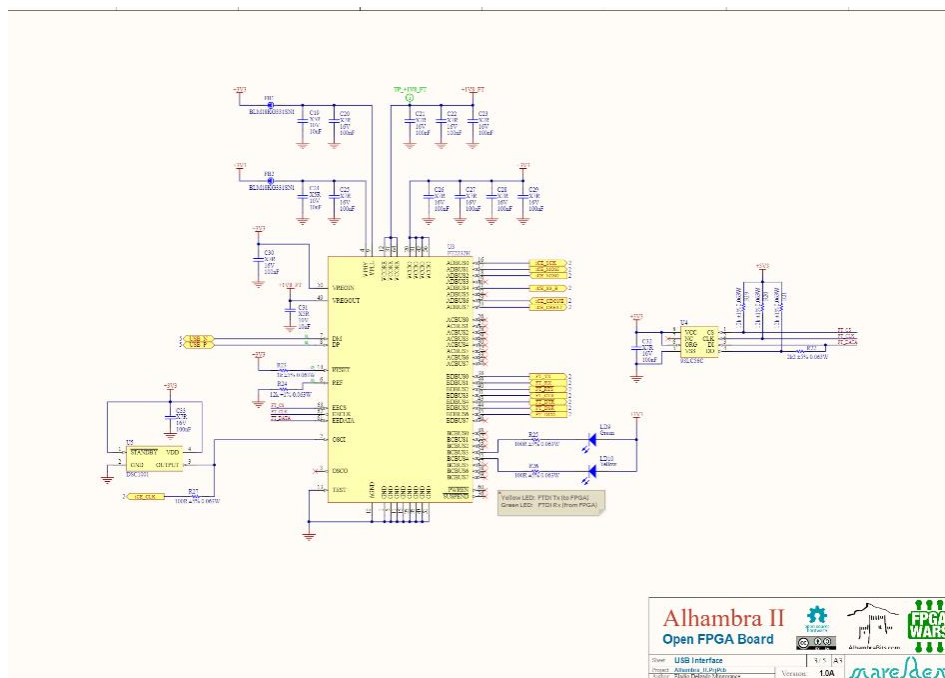


Figura 29 Schematics

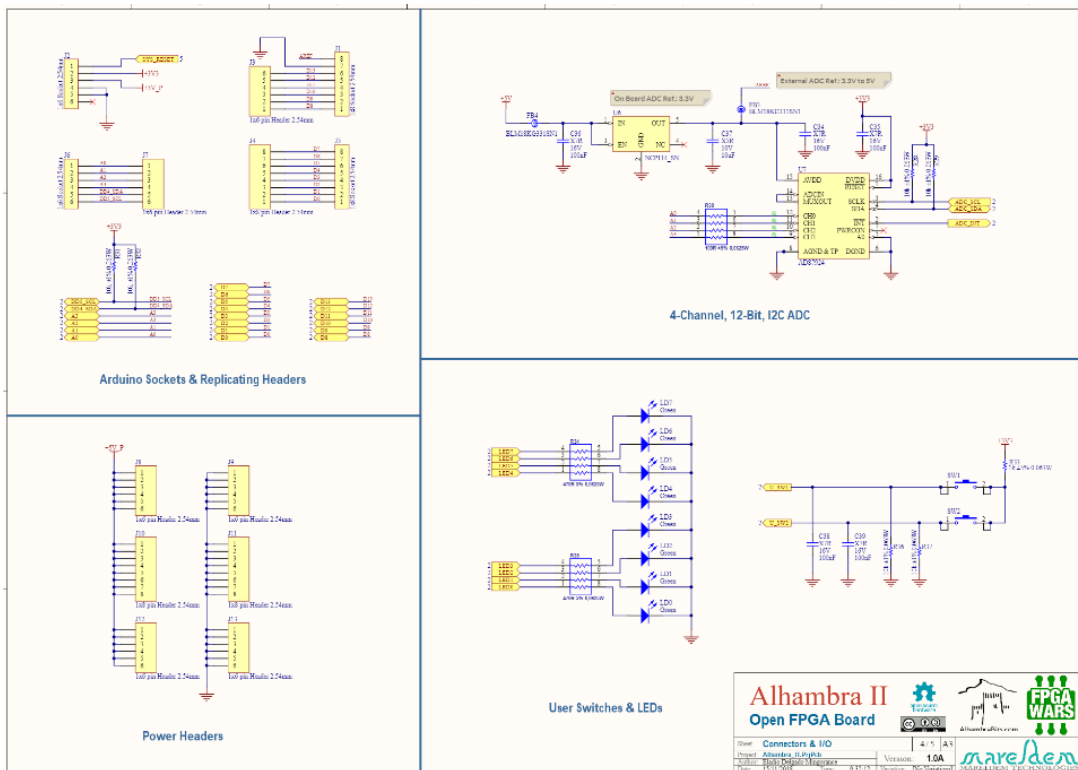


Figure 30 Schematics

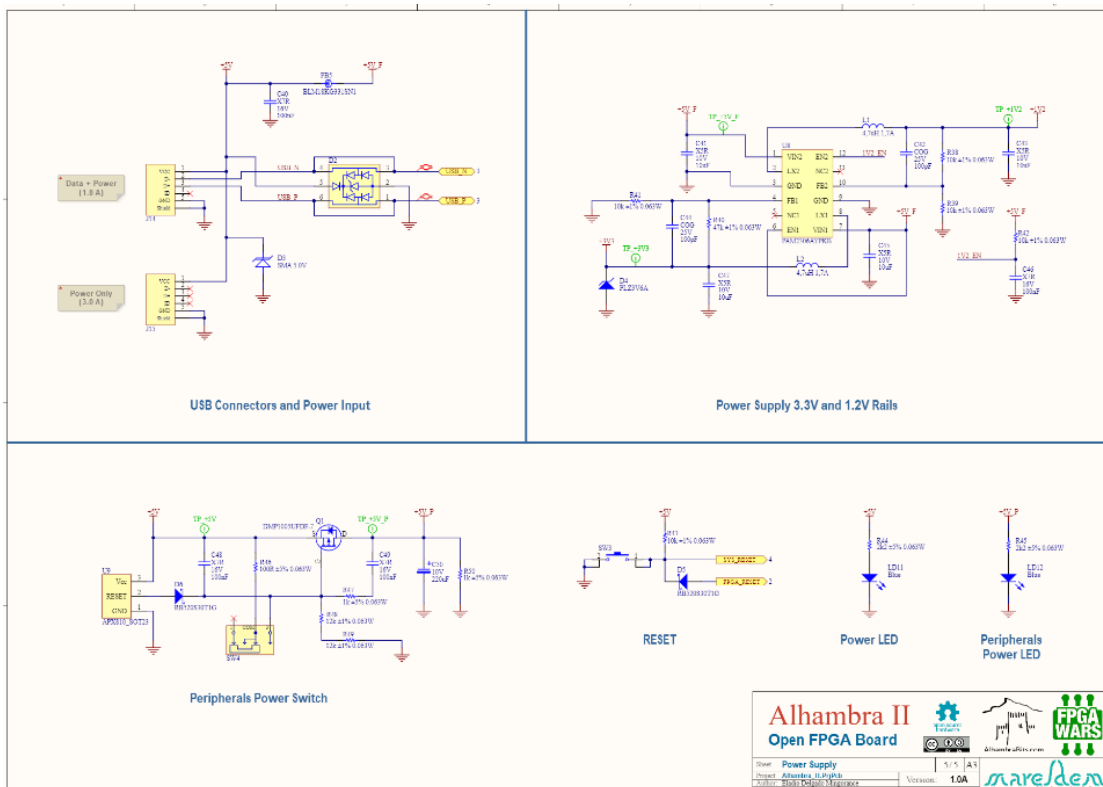


Figure 31 Schematics

## 4.7 PCB

La figura 44 mostra el disseny del PCB de l'Alhambra II on s'observa la distribució dels diferents elements mostrats a la taula 1 (Bill of Material).

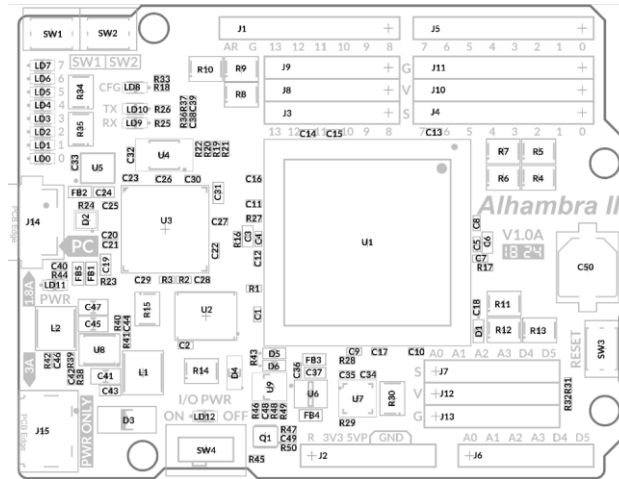


Figura 32 FPGA Alhambra II Disseny PCB

## 5. TRAINING BOARD DE LA UdL

### 5.1 Objectius

En aquest context, al voltant de les FPGAs, seguint el paradigma del hardware lliure i codi obert, es va dissenyar i fabricar a la nostra Escola Politècnica Superior de la UdL una training board de baix cost en la que el nucli és la placa d'entrenament Icezum Alhambra II.

Tot i que l'Icezum Alhambra II disposa de 8 LEDS i 2 “push-button”, això no és suficient per poder fer els tipus de pràctiques que fem, on la majoria de vegades és necessari com a mínim un display per poder visualitzar dades i també els dos polsadors “push-button” són insuficients, ja que es necessiten més de 2 entrades de 1s i 0s.

Es podria haver utilitzat algun dels shields de baix cost, com per exemple els shields multi funció, però la utilització d'aquests shields limita molt la utilització dels pins I/O, ja que les connexions que s'han de fer venen definides pel disseny del hardware.

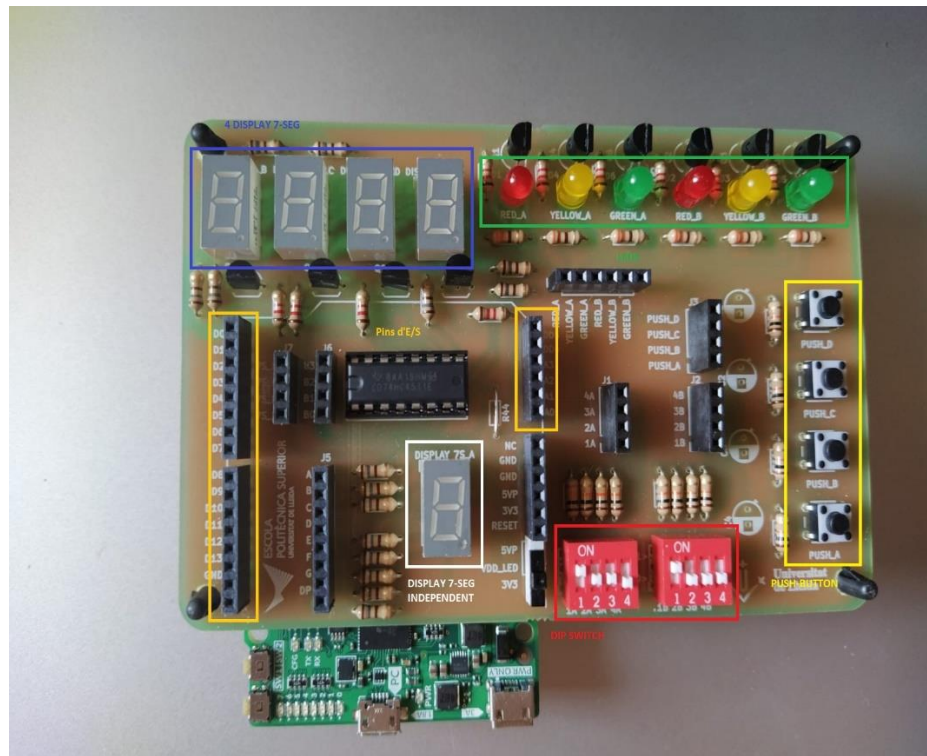
La placa d'entrenament funciona com un shield, és a dir, es connecta als connectors femella de l'Icezum Alhambra II, de manera que queda connectada com una shield damunt de la FPGA.

### 5.2 Característiques

La placa d'entrenament consta de les següents característiques:

- El disseny té unes dimensions de 120mm X 80mm, compatible amb plaques emulsionades de baquelita i/o fibra de vidre comercials de fàcil adquisició i baix cost.
- El disseny ha estat realitzat mitjançant el software KiCad, de forma que els estudiants puguin utilitzar-lo per veure els dissenys i modificar-los.
- El traçat de la PCB ha estat íntegrament en una sola capa.
- Tots els components electrònics utilitzats són through-hole.
- Està formada per:
  - 4 polsadors push-button
  - Dos interruptors DIP Switch de 4 canals
  - Un display de 7-segments controlat per 8 bits independents
  - 4 displays 7-segments controlats per un bus de 4 bits de dades (descodificats a 7-segments mitjançant el circuit integrat CD4511B)
  - 6 LEDs de clors ( 2 vermells, 2 verds i 2 grocs)

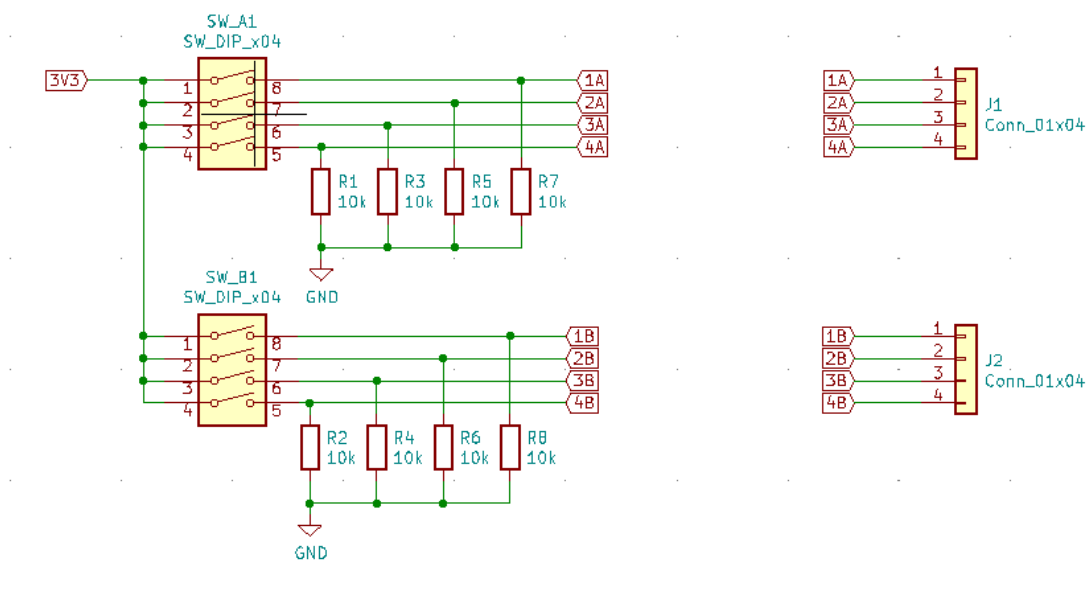
- 20 pins per a I/O
- Pins d'alimentació ( 5V, 3.3V i GND)



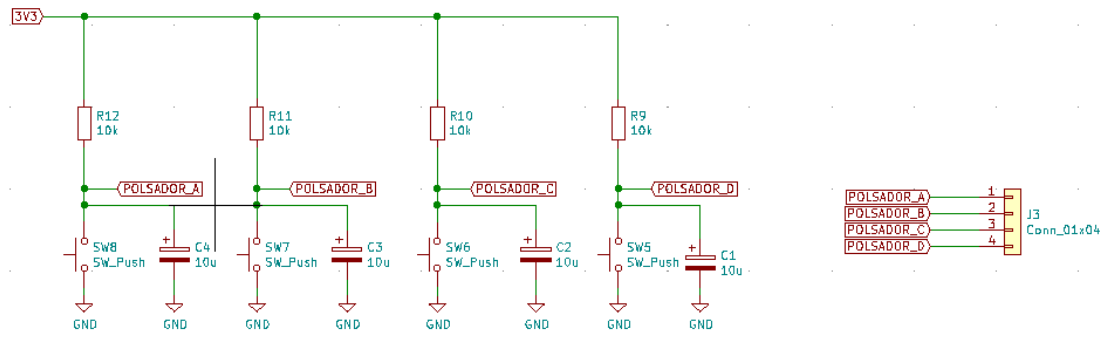
**Figura 33** Training Board

### 5.3 Schematic:

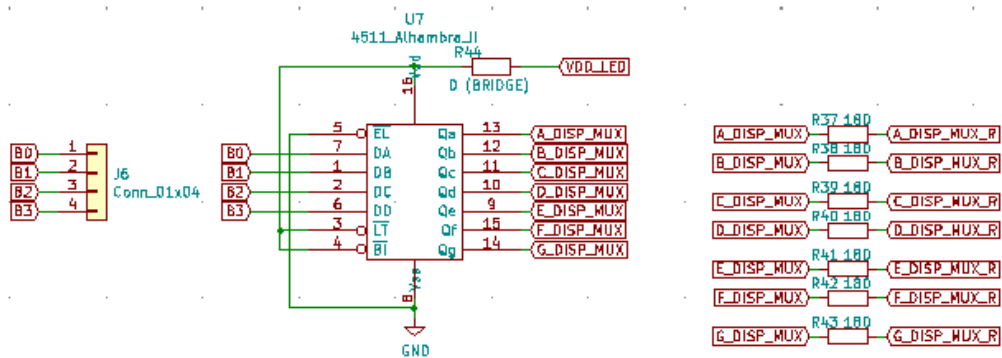
A continuació es mostren els schematics dels diferents components que formen la training board.



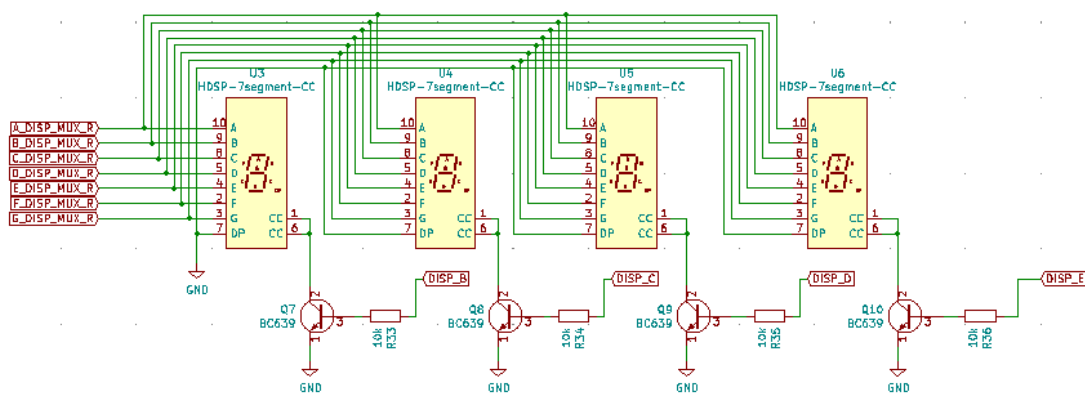
**Figura 34** Polsadors DIP Switch de 4 canals



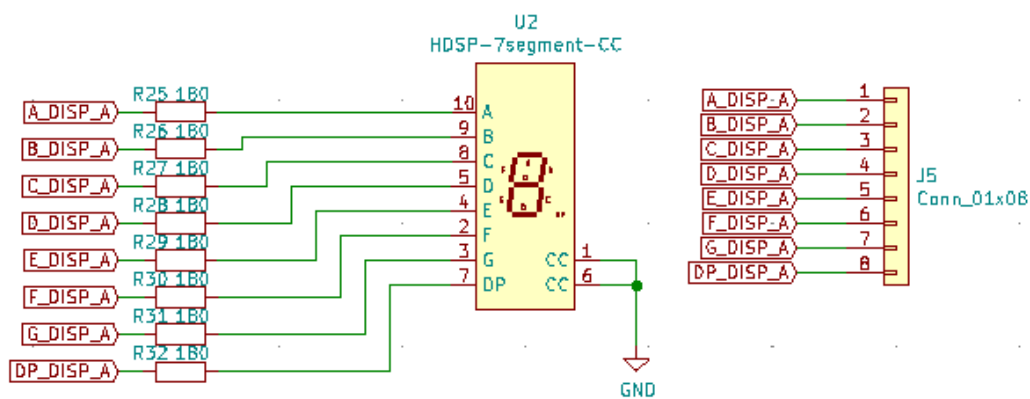
**Figura 35 4** Polsadors Push-button



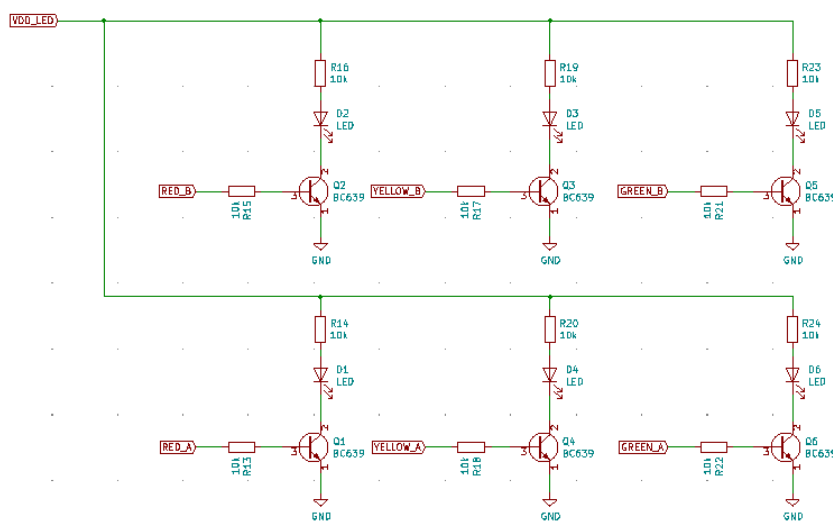
**Figura 36 4** displays 7-segments:



**Figura 37 4** displays 7-segments



**Figura 38** Display 7-segments independent

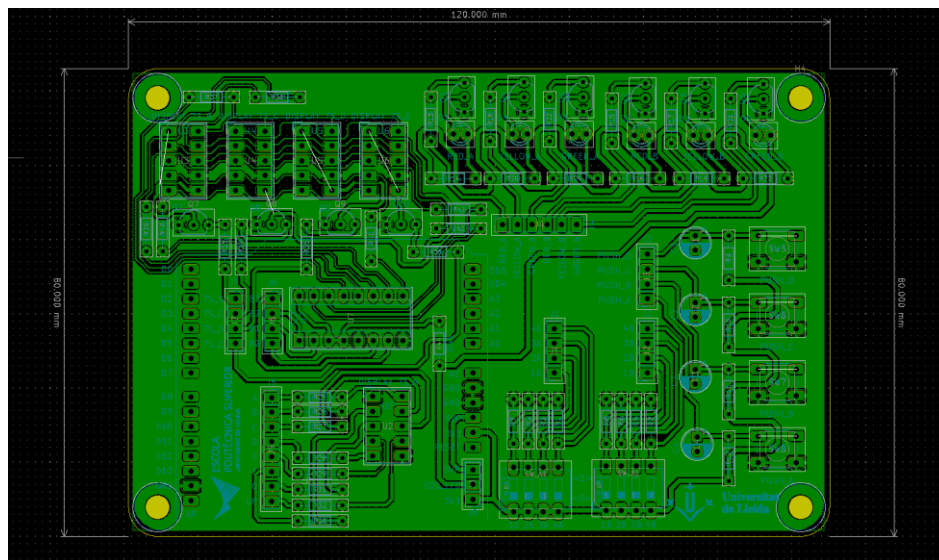


**Figura 39** 6 LED de colors



**Figura 40** 6 LED de colors

## 5.4 PCB:



**Figura 41** Training Board Disseny PCB



## 6. CURS D'APRENTATGE EN LÍNIA

Un dels objectius perseguits amb aquest TFG és complementar la training board amb la creació del curs d'aprenentatge en línia d'introducció a l'electrònica digital mitjançant l'ús de FPGAs lliures, per a que els alumnes puguin realitzar les pràctiques i qualsevol alumne tinguin la possibilitat de fabricar-se la seva pròpia placa d'entrenament mitjançant la fabricació de la PCB per proveïdors externs de molt baix cost, sempre seguint la filosofia del codi obert.

S'ha creat en GitHub un repositori obert, en el qual els alumnes poden accedir a la totalitat de la informació i documentació generada (esquemes, firmware, fitxers gerber de fabricació PCB, Bill of materials etc...).

Així doncs, en Github s'ha creat una wiki on estarien disponibles els arxius i vídeos amb les pràctiques desenvolupades prèviament. En cada pràctica s'especifica quins són els objectius que es volen aconseguir, el plantejament del problema i el procediment per dur-la a terme.

### 6.1 Github

Github, Inc. és una companyia amb seu als Estats Units que proporciona allotjament per al control de versions de desenvolupament de software utilitzant Git.

El desenvolupament de la plataforma GitHub va començar el 19 d'octubre de 2007.

El lloc web va ser llançat a l'abril de 2008 per Tom Preston, Chris Wanstrath i Scott Chacon, després d'haver estat durant uns quants mesos una versió beta disponible. L'any 2018 es va convertir en una subsidiària de Microsoft per un preu de 7.5 billons de dòlars.

Es pot accedir i manipular projectes en GitHub utilitzant la interfàç de línia de comandes estàndard de Git i permet als usuaris registrats i no registrats poder navegar per tots els repositoris públics del lloc.

Els comptes gratuïts de GitHub s'utilitzen comunament per poder allotjar projectes de codi obert. GitHub funciona d'una forma bastant similar a les xarxes socials, amb seguidors, wikis, etc.

Un usuari s'ha de crear un compte dins de GitHub per a poder contribuir amb contingut dins del lloc web, però no fa falta estar registrat per poder visualitzar i descarregar els projectes d'altres usuaris. El software fonamental que manté GitHub és el propi Git, escrit per Linus Torvalds, creador de Linux.

GitHub s'utilitza bàsicament per emmagatzemar codi però a més, admet una sèrie de formats i característiques:

- Documentació
- Seguiment de problemes
- Wikis
- Gràfics
- Emojis
- Etc

És per aquest motiu pel qual s'ha escollit GitHub com el lloc on poder emmagatzemar tota la informació sobre les pràctiques a realitzar amb la placa d'entrenament i on aniran tots els repositoris que contindran els schematics i fitxers Gerber i Bill Of Material.

## 6.2 Pràctiques proposades

### 6.2.1 Pràctica 1

#### ➤ Enunciat:

Mostrar un únic dígit (4 bits en binari) a través del sistema de 4 displays 7-segments multiplexats de la training FPGA-UdL amb selecció de velocitat de refresc/multiplexació dels displays.

#### ➤ Connexions a la placa de training FPGA-UdL:

#### ➤ Introducció del dígit:

Per poder introduir el dígit, s'ha de fer mitjançant la placa de training FPGA-UDL. Per això, utilitzem el DIP Switch B, de forma que el dígit en binari (x3,x2,x1,x0) quedarà en funció del valor que nosaltres assignem als 4 polsadors del DIP Switch B, quedant d'aquesta forma (4B,3B,2B,1B). Ara que ja tenim configurada la introducció del dígit, falta connectar els pins (4B,3B,2B,1B) amb les entrades que volem utilitzar a la FPGA, que en el nostre cas són (D4,D5,D6,D7).

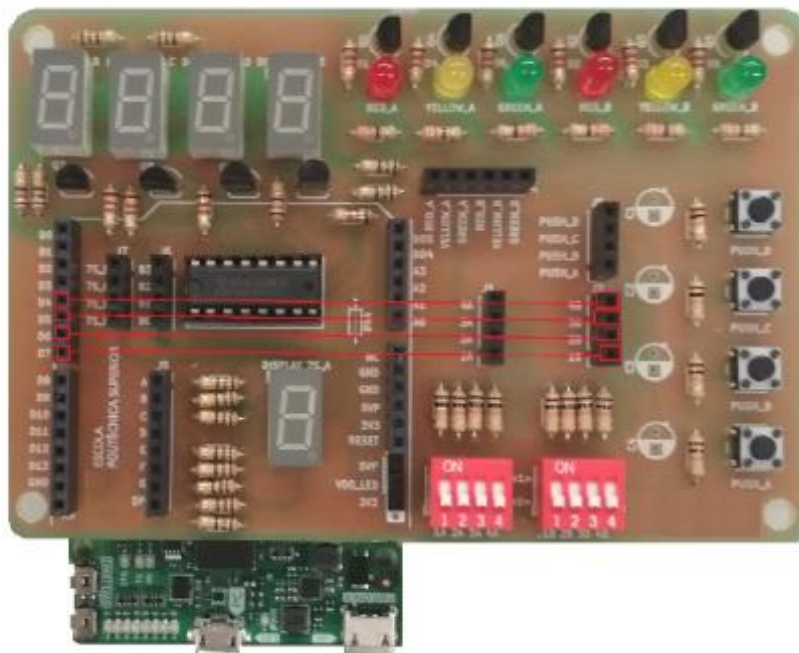
Les connexions es fan tal i com indica la següent imatge:

D4 <-- 4B

D5 <-- 3B

D6 <-- 2B

D7 <-- 1B



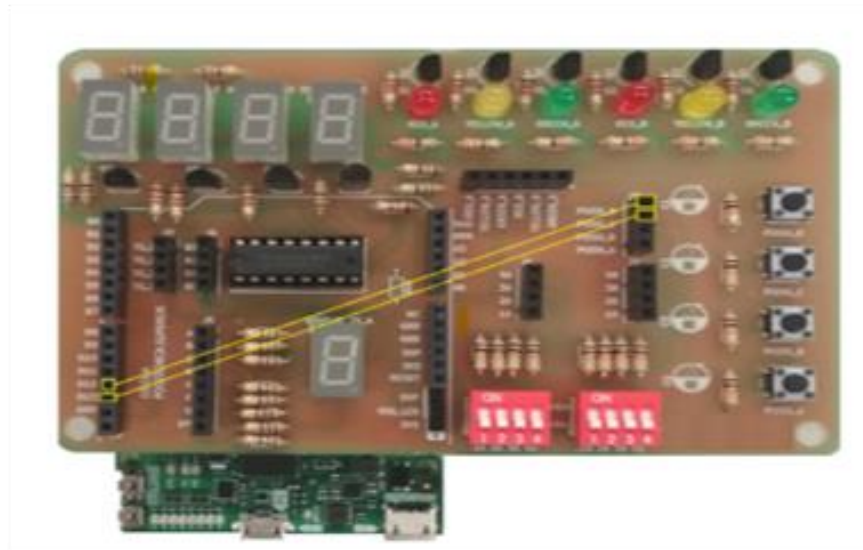
**Figura 42** Introducció del Digit

➤ **Connexions per la selecció del clock:**

Per seleccionar les diferents velocitats del rellotge utilitzarem un nombre de 2 bits, que estarà format pels estats en els que estiguin els push\_buttons D i C. Aquests dos push\_Button podran variar entre l'estat 0 (ningú l'està polsant) i l'estat 1 (l'estan polsant) i formar quatre possibles combinacions, que ens indicaran quina velocitat volem seleccionar. Un cop tenim connectat els dos push\_Buttons, falta portar aquest senyal de control als pins que nosaltres em indicat que utilitzarem a la nostra FPGA, en aquest cas els pins D12 , D13 i els connectarem amb els push\_button tal i com indica la següent imatge.

D12 <-- PUSH\_D

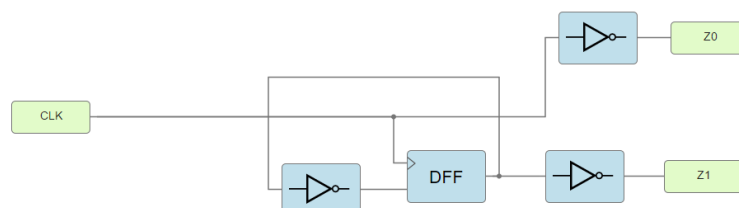
D13 <-- PUSH\_C



**Figura 43** Selecció del Rellotge

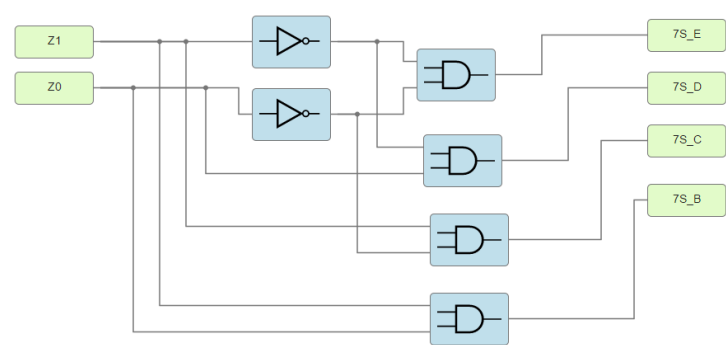
➤ **Selecció del display a mostrar:**

Ja que en la pràctica ens demana que anem mostrant el dígit pels diferents displays, haurem de poder controlar quin display volem que estigui actiu. Per poder fer això, la FPGA disposa de 4 entrades (7S\_B,7S\_C,7S\_D,7S\_E) on cadascuna d'aquestes entrades controla un transistor que activa o desactiva el seu corresponent display. Per anar variant entre els 4 displays, en aquesta pràctica s'ha optat per controlar quin display s'activarà mitjançant un nombre de 2 bits, que estarà generat per un flip-flop D. Aquest nombre podrà ser entre 0 i 3, sent 00-01-10-11 en binari. Aquest nombre anirà canviant entre els seus possibles valors de forma seqüencial amb cada pulsació del clock i a cada possible valor anirà assignat un display que haurà d'encendre. Per generar el número de selecció de display, s'utilitzarà un flip-flop D, amb el clock que nosaltres hàgim seleccionat a l'apartat anterior. El seu esquema és el següent:



**Figura 44** Flip-Flop D

Aquest flip-flop està dissenyat per a que vagi traient com a sortida la seqüència 00-01-10-11 amb la velocitat del clock introduït. Un cop tinguem el número de selecció (Z1,Z0) de display en binari, s'utilitzaran una sèrie de portes lògiques per activar segons cada estat de la seqüència un display diferent tal i com s'indica a la següent imatge:



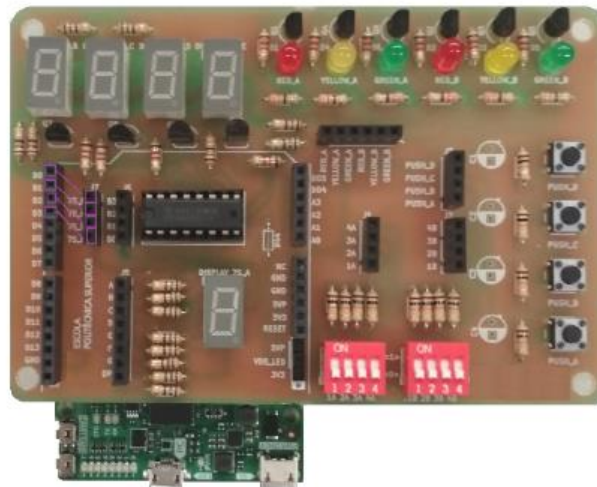
**Figura 45** Sequència Displays

Quan el número equival al 00 (Z1=0,Z0=0), indicarà que volem activar l'entrada 7S\_E del display, per tant necessitarem 2 portes NOT per transformar el 00 a 11 i activar el transistor amb una porta AND.

Z1	Z0	DISPLAY
0	0	7S_E
0	1	7S_D
1	0	7S_C
1	1	7S_B

**Taula 2** Selecció del Display a mostrar

Les connexions a la FPGA queden de la següent forma:

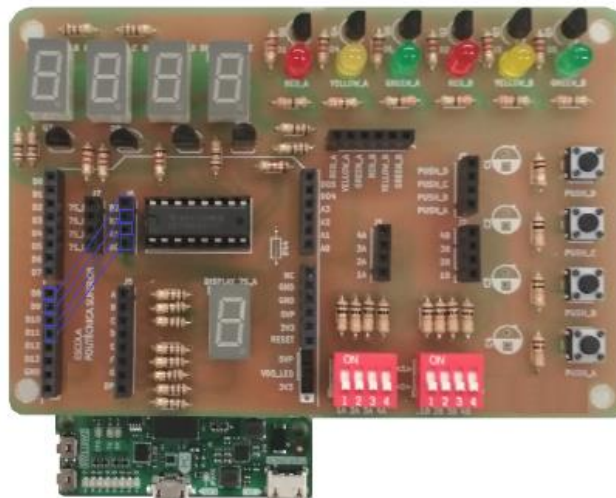


**Figura 46** Connexions a la Training Board

➤ **Connexions als displays:**

La sortida  $O[3..0]$  es treu pels pins D8, D9, D10 i D11 de la FPGA i llavors es connecten a l'entrada del xip 4511

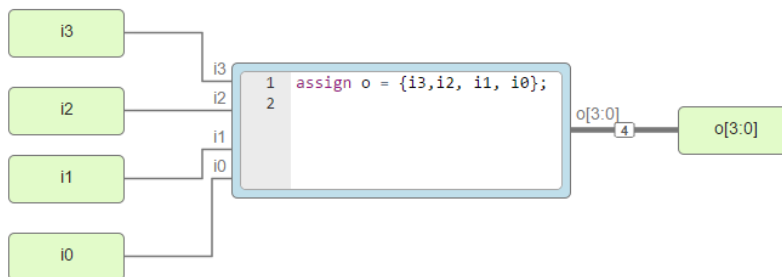
- D8 --> B0
- D9 --> B1
- D10 --> B2
- D11 --> B3



**Figura 47** Connexions als Displays

### ➤ Realització de la pràctica:

En primer lloc, per evitar haver de realitzar tantes connexions, treballarem amb un bus de 4 bits que contindrà el dígit que nosaltres em introduït mitjançant la FPGA. Aquest bus de 4 bits s'obté de la llibreria amb el nom "Agregador Bus", però es pot dissenyar manualment mitjançant codi Verilog. Aquest agregador Bus està programat de la següent forma:



**Figura 48** Agregador Bus

### ➤ Selecció del clock:

En aquesta pràctica ens demana mostrar el dígit amb 4 velocitats diferents, és a dir, que em d'utilitzar 4 clocks diferents i anar variant entre ells. De totes les opcions que hi ha per fer-ho, en aquesta pràctica s'ha optat per seleccionar les diferents velocitats mitjançant un multiplexor de 4 a 1. Per fer-ho, es connecta els 4 clocks a les 4 possibles línies de sortida del multiplexor i es selecciona quina d'aquestes línies es vol treure pel multiplexor mitjançant els dos bits de selecció que s'introdueixen mitjançant la connexió amb la FPGA-UDL, amb els pins (D13,D12). D'aquesta forma la sortida quedarà en funció del nombre binari "selecció"= (seleccio1, seleccio0).

seleccio1	seleccio0	Sortida
0	0	CLK1
0	1	CLK2
1	0	CLK3
1	1	

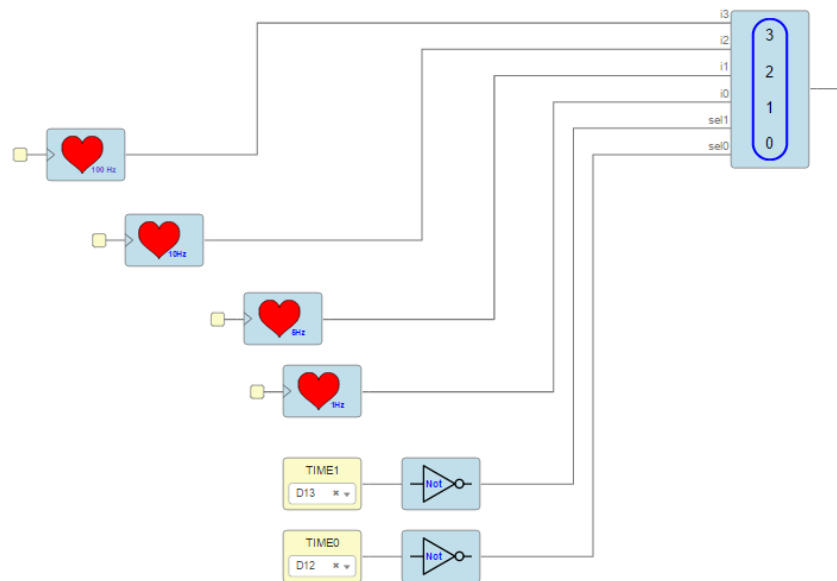
**Taula 3** Taula de la veritat selecció del clock:

CLKx	Ix
CLK1	I0
CLK2	I1
CLK3	I2
CLK4	I3

**Taula 4** Assignació de clocks a les entrades del multiplexor

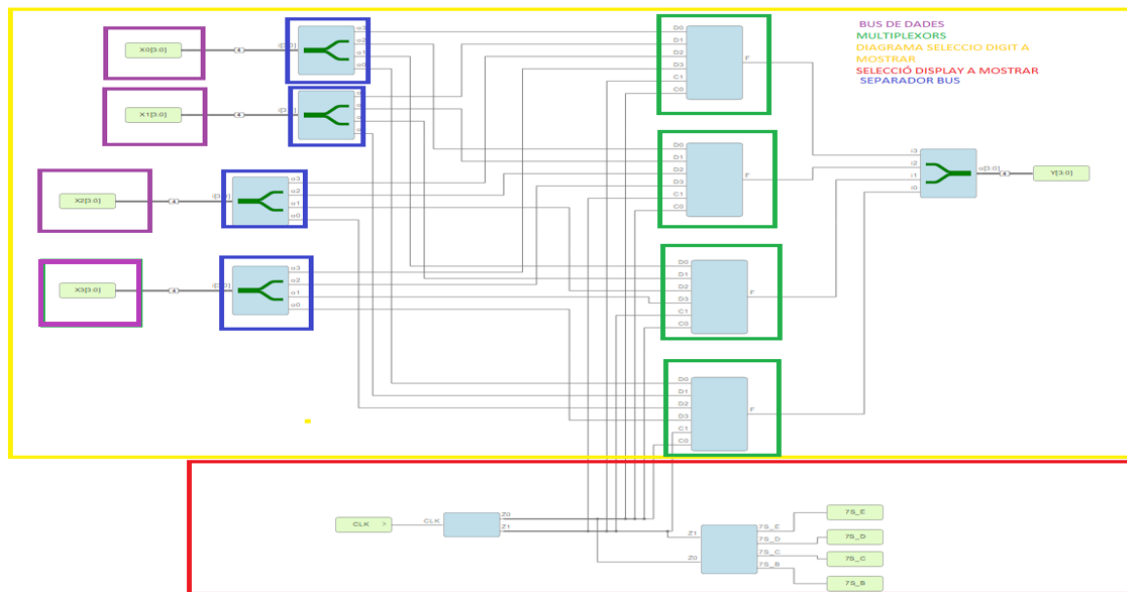


➤ **Diagrama connexions Icestudio**



**Figura 49** Realització IceStudio

La part vermella pertany al diagrama de generació de les variables de control Z1,Z0 i el control per seleccionar quin display s'ha d'il·luminar. La part groga pertany al procés per seleccionar el dígit a introduir. El bloc de la imatge està pensat per poder introduir 4 dígits de 4 números binaris cadascun i poder anar-los seleccionant mitjançant les variables de Z1,Z0. Com a l'enunciat ens indica que només hem d'introduir un únic dígit de 4 bits, només estarà en funcionament un únic bus de dades, el XO[3,0]. Aquest dígit en binari entrarà com a I0 en els 4 multiplexors, de forma que a cada multiplexor hi haurà un únic dígit binari, és a dir, x3-x2-x1-x0 segons el multiplexor que toqui. Amb la senyal de selecció Z1,Z0 com a selecció dels multiplexors, aquests aniran traient els dígits que corresponguin segons quin valor de selecció hi ha en aquell moment. Ja que només em introduït un dígit, quan selecció valgui un número diferent de 00, els displays mostraran 0, ja que no hi ha cap altre número connectat a les altres Ix del multiplexor. Per tant, quedarà així:

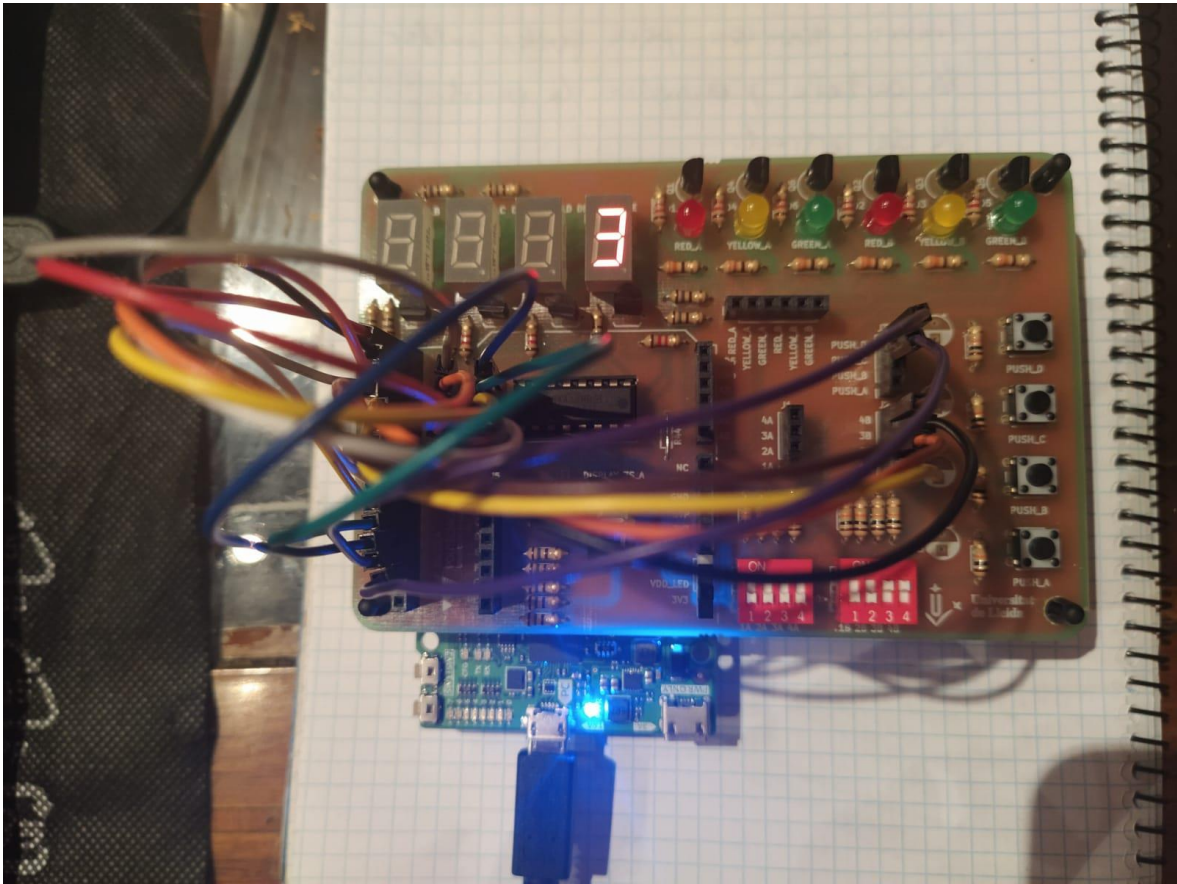


**Figura 50** Realització IceStudio

Z1	Z0	Sortida Mux
0	0	x3,x3,x2,x0
0	1	0,0,0,0
1	0	0,0,0,0
1	1	0,0,0,0

**Taula 5** Sortida Multiplexor

➤ **Funcionament Pràctica 1:**



**Figura 51** Funcionament Pràctica 1

## 6.2.2 Pràctica 2

➤ **Enunciat Pràctica 2:**

Mostrar dos dígits (4 bits en binari cadascú) a través del sistema de 4 displays 7-segments multiplexats de la training FPGA-UdL amb velocitat de refresc/multiplexació dels displays fixa (100 Hz).

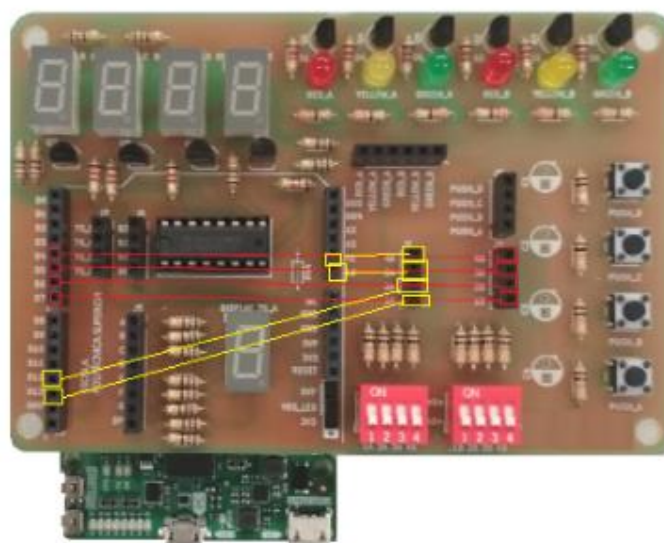
➤ **Connexions a la placa de training FPGA-UdL:**

➤ **Introducció dels dígits:**

Novament, haurem d'introduir els dos dígits mitjançant la placa FPGA-UdL. En aquesta ocasió haurem d'utilitzar els dos DIP Switch, ja que necessitem 4 bits per cada número. Assignarem el primer dígit al DIP Switch B tal com hem fet a la pràctica anterior i el segon dígit al DIP Switch A. Les connexions quedaran de la forma següent:

Primer Dígit (Vermell)	Segon Dígit (Groc)
D4 <-- 4B	D12 <-- 2A
D5 <-- 3B	D13 <-- 1A
D6 <-- 2B	DD1 <-- 4A
D7 <-- 1B	DD0 <-- 3A

**Taula 6** Introducció dels dos dígets.

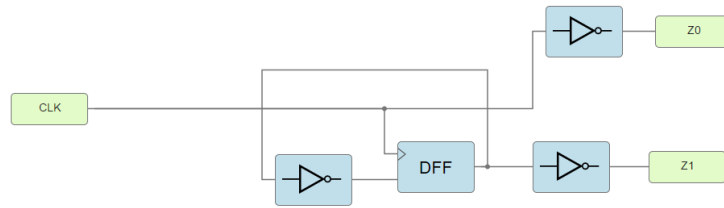


**Figura 52** Introducció dels dos dígets

➤ **Selecció del display a mostrar:**

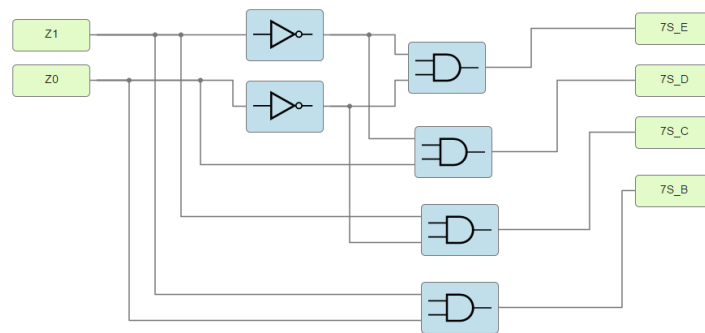
El funcionament d'aquesta part serà el mateix que el realitzat a la pràctica 1, amb l'única diferència que en aquesta pràctica tenim un temps de clock fixe de 1000 Hz. Al tenir un senyal de clock que fa que variïn els senyals de control (Z1,Z0) tant ràpidament provocarà que es vegin els dos dígets als dos displays de forma continua, si el senyal de clock fos inferior, com per exemple 1Hz, el que veuríem a la placa entrenadora és que els dos dígets s'anirien mostrant seqüencialment i mai es veurien els dos a la vegada. Al tenir un clock elevat provoquem que es doni l'efecte òptic i pensem que s'estan mostrant els dos dígets a la vegada.

Per generar el número de selecció de display, s'utilitzarà un flip-flop D, amb el clock de 100 Hz. El seu esquema és aquest:



**Figura 53** Flip-Flop D

Aquest flip-flop està dissenyat per a que vagi traient com a sortida la seqüència 00-01-10-11 amb la velocitat del clock de 100 Hz. Un cop tinguem el numero de selecció (Z1,Z0) de display en binari, s'utilitzaran una sèrie de portes lògiques per activar segons cada estat de la seqüència un display diferent tal i com indica la següent imatge:



**Figura 54** Sequència Displays

Quan el número equival a 00 ( $Z1=0, Z0=0$ ), indicarà que volem activar l'entrada 7S\_E del display, per tant necessitarem 2 portes NOT per a transformar el 00 a 11 i activar el transistor amb una porta AND.

Z1	Z0	DISPLAY
0	0	7S_E
0	1	7S_D
1	0	7S_C
1	1	7S_B

**Taula 7** Selecció del Display a mostrar

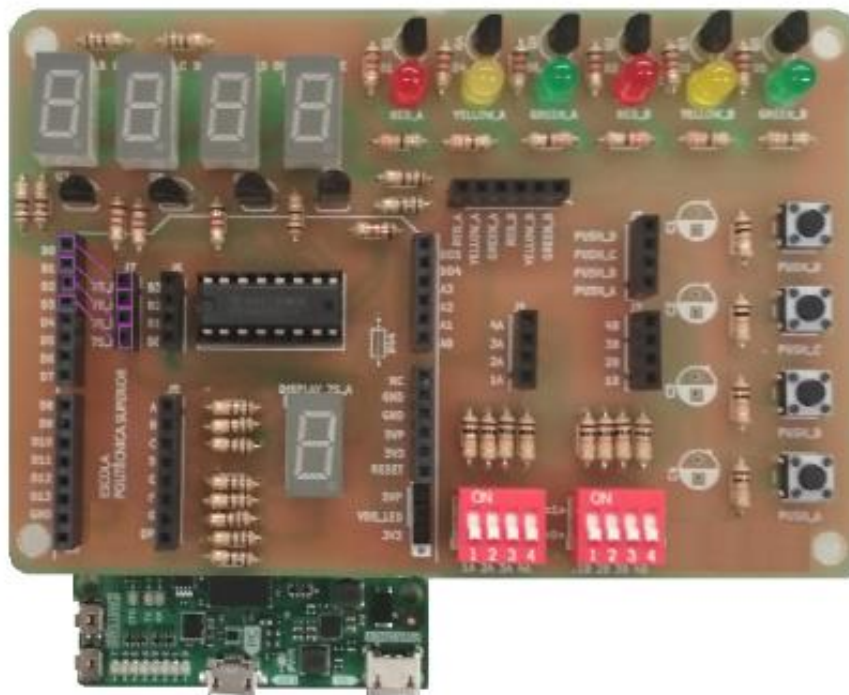
Les connexions a la FPGA queden de la següent forma:

➤ **Connexions als displays:**

La sortida O[3..0] es treu pels pins D8, D9, D10 i D11 de la FPGA i llavors es connecten a l'entrada del xip 4511

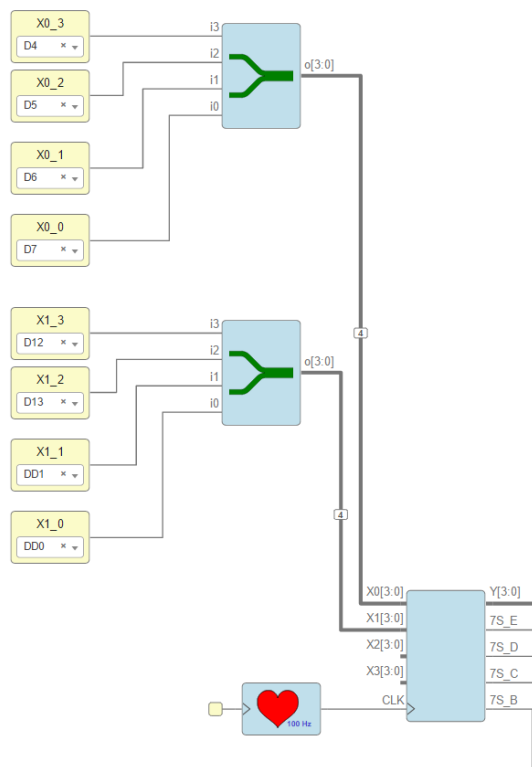
D8	→	B0
D9	→	B1
D10	→	B2
D11	→	B3

**Taula 8** Connexions als displays de la training board



**Figura 55** Connexions als Displays

➤ **Diagrama connexions introducció 2 dígits Icestudio:**



**Figura 56** Realització Icestudio

➤ **Diagrama connexions Icestudio**

La part vermella pertany al diagrama de generació de les variables de control Z1,Z0 i el control per seleccionar quin display s'ha d'il·luminar. La part groga pertany al procés per seleccionar el dígit a introduir. El bloc de la imatge està pensat per poder introduir 4 dígits de 4 números binaris cada un i poder anar-los seleccionant mitjançant les variables de Z1,Z0. Com l'enunciat ens indica que només hem d'introduït un únic dígit de 4 bits, només estarà en funcionament un únic bus de dades, el XO[3,0]. Aquest dígit en binari entrarà com a I0 en els 4 multiplexors, de forma que a cada multiplexor hi haurà un únic dígit binari, és a dir, x3-x2-x1-x0 segons el multiplexor que toqui. Amb el senyal de selecció Z1,Z0 com a selecció dels multiplexors, aquests aniran traient els dígits que corresponguin segons quin valor de selecció hi ha en aquell moment, ja que només em



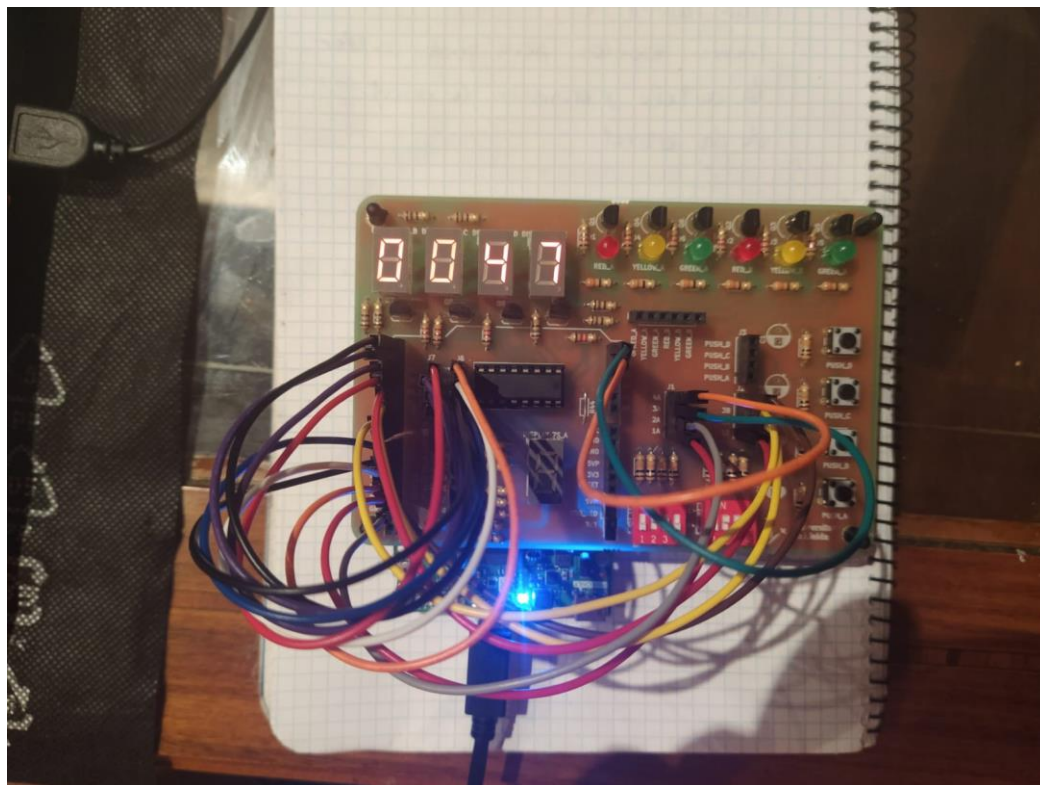
introduït un dígit. Quan la selecció valgui un número diferent de 00, els displays mostraran 0, ja que no hi ha cap altre número connectat a les altres lx del multiplexor.

Per tant, quedarà així:

Z1	Z0	Sortida Mux
0	0	x3,x3,x2,x0
0	1	0,0,0,0
1	0	0,0,0,0
1	1	0,0,0,0

**Taula 9** Sortida Multiplexor

➤ **Funcionament Pràctica 2:**



**Figura 57** Funcionament Pràctica 2

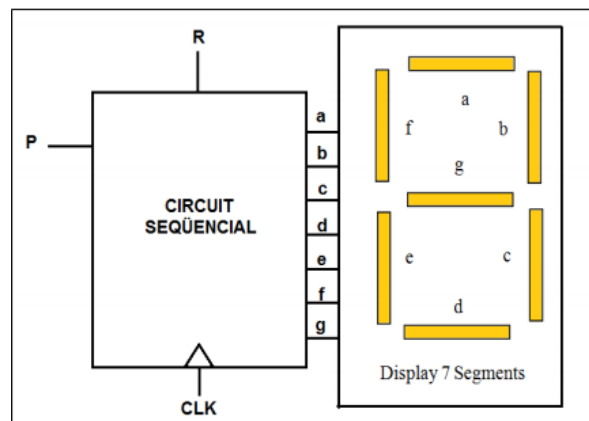


### 6.2.3 Pràctica 3

#### ➤ Enunciat:

L'objectiu d'aquesta pràctica és dissenyar un circuit seqüencial amb entrades i sortides utilitzant flip-flops.

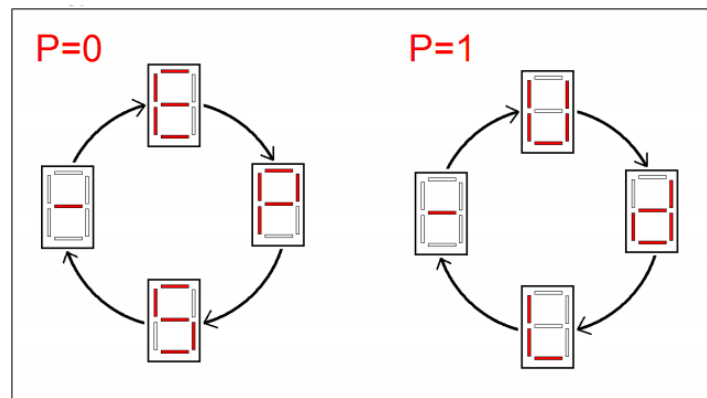
ESTUDI PREVI: Es vol dissenyar un circuit seqüencial que controli els caràcters a visualitzar en un display de 7 segments.



**Figura 58** Enunciat Pràctica 3

El circuit té un funcionament síncron amb una entrada de rellotge CLK i una entrada de control P que governa la seqüència de caràcters a visualitzar en el display de 7 segments.

- Quan l'entrada P val 0 es visualitzarà la seqüència EPS-EPS-EPS-EPS-.....
- Quan l'entrada P val 1 apareixerà al display la seqüència de caràcters UdL-UdL-UdL-.....



**Figura 59** Enunciat Pràctica 3

El sistema també disposa d'una entrada R (reset) asíncrona que permet aturar la seqüència en qualsevol instant.

➤ **Realització de la pràctica:**

Aquesta pràctica té diverses formes de dur-se a terme. En aquest cas la forma de generar les lletres en seqüència E-P-S o U-D-L es fa mitjançant 4 possibles estats, que es generaran de forma seqüencial mitjançant dos flip-flops JK. Aquests flip-flops generaran 4 possibles valors, que seran 00-01-10-11. Amb aquests 4 possibles estats, mitjançant Karnaugh podrem activar els pins indicats per formar a cada estat la lletra que volem mitjançant portes lògiques.

Les taules de la veritat seran:

P	Q1	Q0	a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	1
0	0	1	1	0	0	1	1	1	0
0	1	0	1	1	0	0	1	1	1
0	1	1	1	0	1	1	0	1	1
1	0	0	0	0	0	0	0	0	1
1	0	1	0	1	1	1	1	1	0
1	1	0	1	1	1	1	1	1	0
1	1	1	0	0	1	1	1	1	0

**Taula 10** Taula de la veritat

Fent Karnaugh i posant les entrades del display 7-segments en funció de P, Q1, Q0 ens dona:

$$A = (Q_0' \cdot P') + (P' \cdot Q_1')$$

$$B = (Q_1' \cdot Q_0) + (P \cdot Q_1')$$

$$C = (Q_1 \cdot Q_0') \cdot P' + (P \cdot Q_1')$$

$$D = Q_0' + (P \cdot Q_1')$$

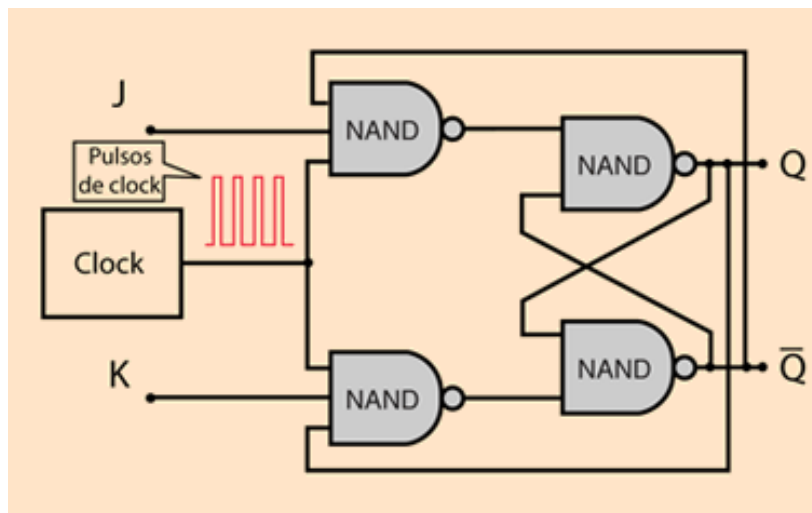
$$E = (P \cdot Q_0') + Q_1'$$

$$F = Q_0' + (P' \cdot Q_1')$$

$$G = Q_0 + P'$$

### FLIP-FLOP PROBLEMES DE CARRERA:

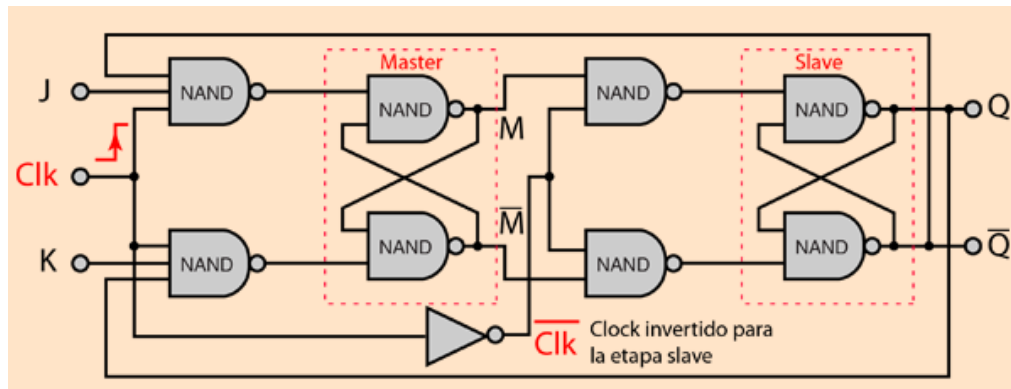
Una versió simplificada del flip-flop jk seria la següent:



**Figura 60** Flip-Flop D

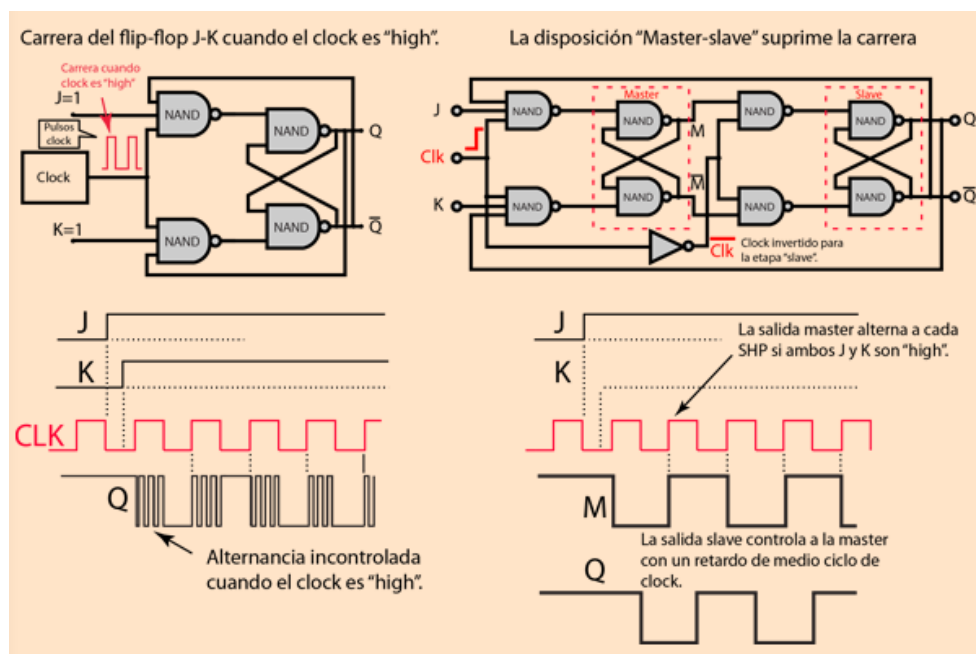
Es pot observar com les sortides es retro alimenten a les portes NAND habilitades. Això és el que dóna pas a l'acció de commutació quan  $J=K=1$ . Si bé aquesta configuració amb quatre portes NAND funciona en principi, ens adonem que sorgeixen problemes amb el temps (timing). Per aquesta versió del flip-flop jk, la commutació s'activaria en qualsevol moment en que arribés un flanc de pujada del clock del sistema, és a dir, quan el clock valgués 1. La velocitat d'activació vindria donada pel retard de propagació al llarg del propi sistema. Per tant, el valor de sortida en qualsevol moment no seria

predicible per l'estat del clock. El puls de "timing" ha d'ésser molt curt, perquè un canvi en el valor de Q abans que s'acabi provocaria una oscil·lació en el circuit que es denomina "carrera". Per sort, aquesta commutació no controlada es pot suprimir mitjançant una configuració del flip-flop, que es denomina "Master-Slave"



**Figura 61** Flip-Flop D en configuració Master-Slave

En aquesta nova disposició es retarda la transmissió del valor J a la sortida amb un valor de la meitat del clock i no es retroalimenta immediatament a la entrada, ocasionant que no es produeixi aquest efecte.

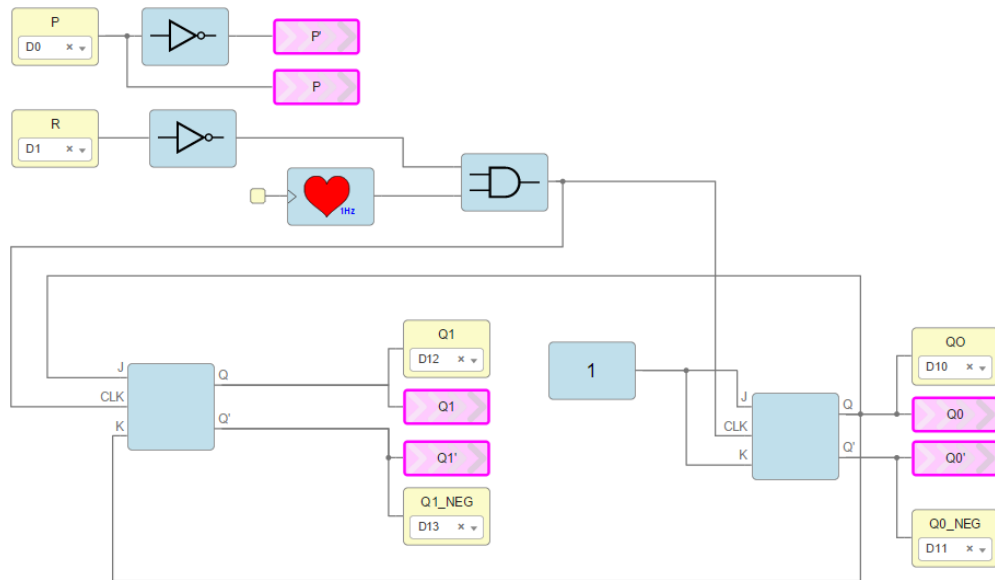


**Figura 62** Comparació Transmissió de la Senyal

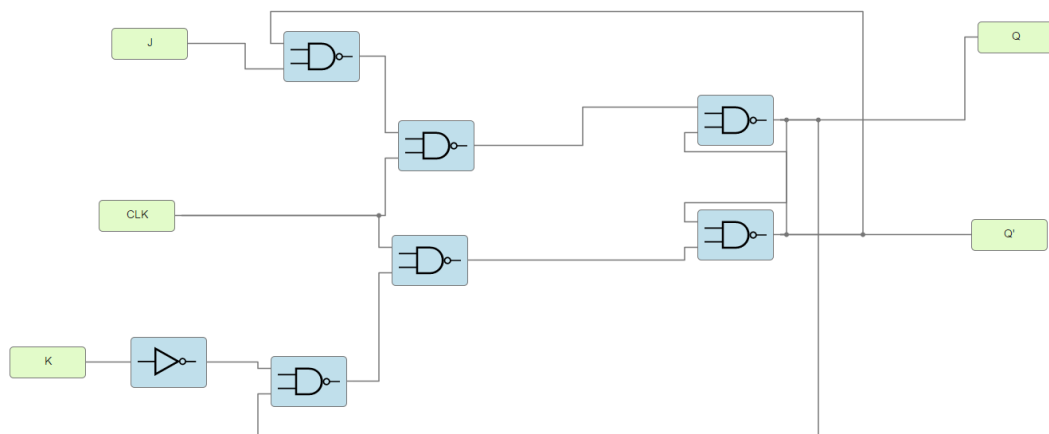
➤ **Assignació de P i R:**

Introduïm mitjançant els Dip Switch els valors P i R  $D0 \leftarrow 1A$   $D1 \leftarrow 1B$

➤ **Esquema del flip-flop:**



**Figura 63** Realització Flip-Flop D iCEstudio



**Figura 64** Configuració Master-Slave

La generació dels 4 estats la formarem mitjançant la unió dels dos flaps JK. El primer flip-flop tindrà tant la J com la K connectades a un valor de 1 i el segon les tindrà connectades al valor de Q0. Per controlar que quan es premi el boto R el sistema es quedi bloquejat, es controla la senyal del clock connectada mitjançant una porta AND a

la senyal R negada, de forma que quan R es trobi en estat actiu, el flip-flop deixi de rebre el clock i es quedi bloquejat a la lletra que tenia assignada en aquell moment.

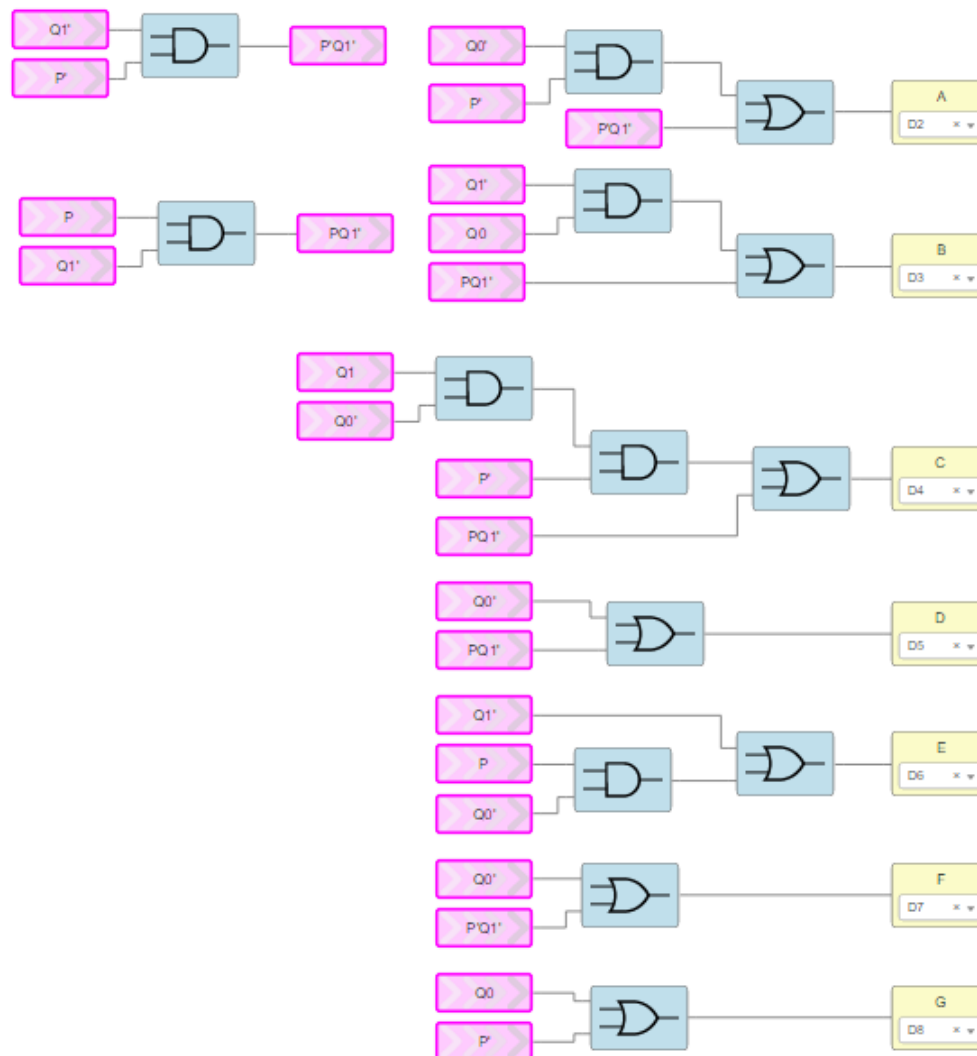
➤ **Assignació de les sortides (a-b-c-d-e-f-g) als pins de l'entrenador:**

Mostrem caràcters a través del display 7-segments i estat dels Flip-Flops pels LEDS grocs i verds

D2 ➔ A
D3 ➔ B
D4 ➔ C
D5 ➔ D
D6 ➔ E
D7 ➔ F
D8 ➔ G
D9 ➔ YELLOW_B
D10 ➔ GREEN_B
D11 ➔ YELLOW_A
D12 ➔ YELLOW_A

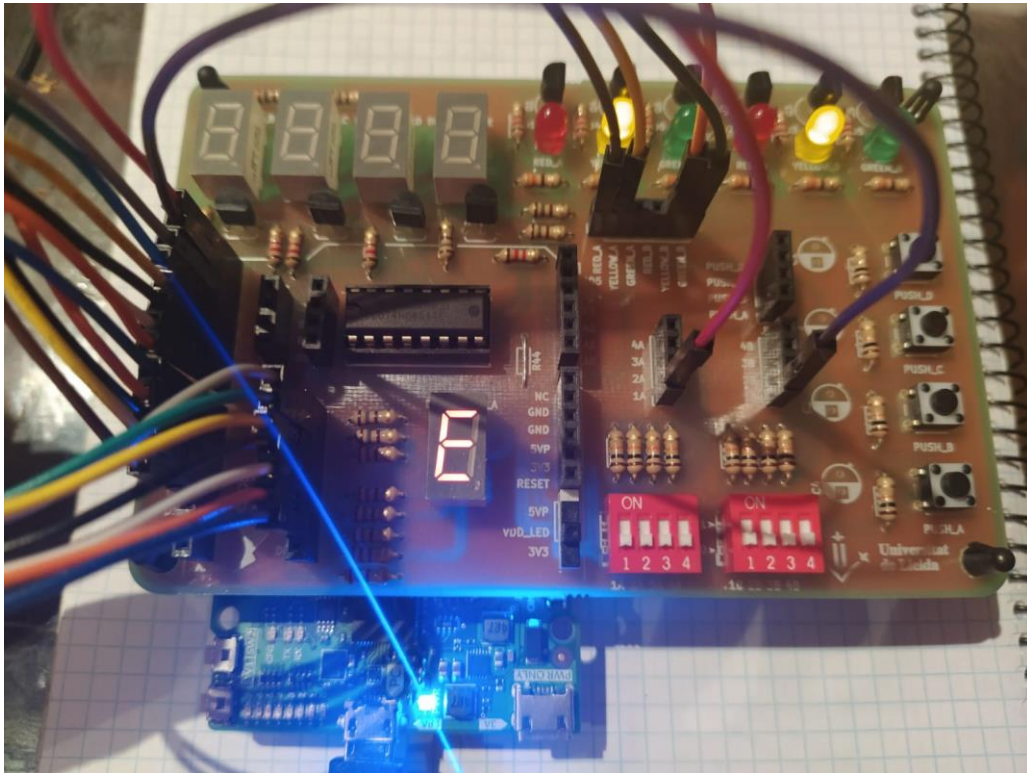
**Taula 11** Assignació de sortides

➤ Realització Karnaugh:

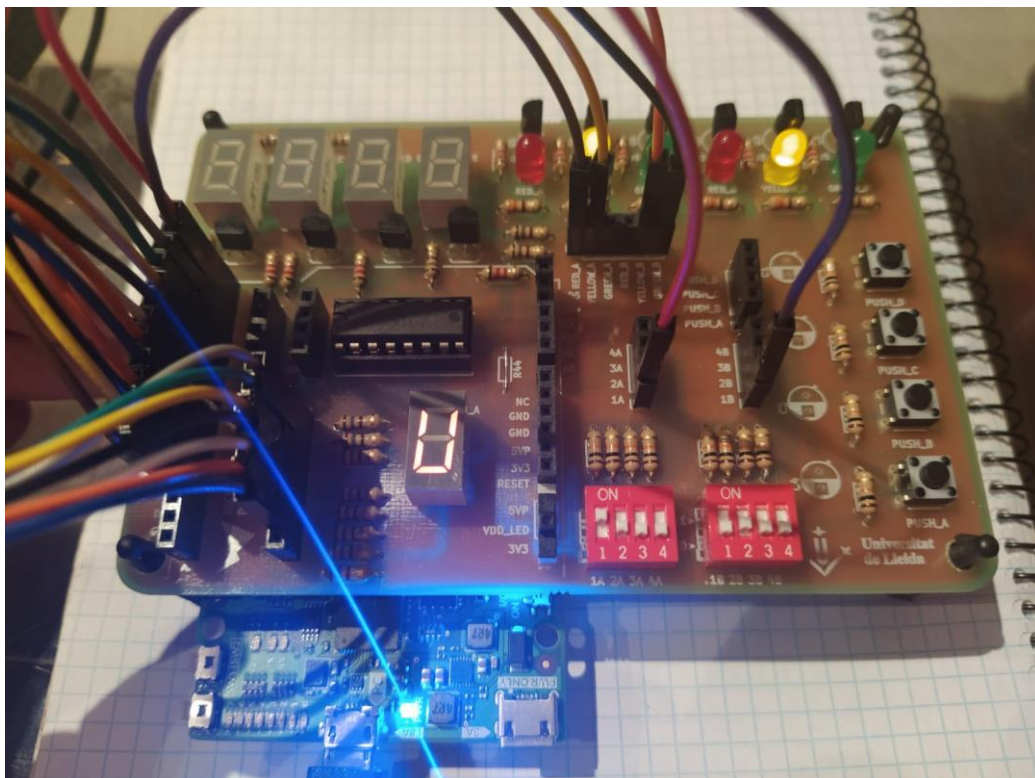


**Figura 65** Karnaugh

➤ **Funcionament Pràctica 3:**



**Figura 66** Funcionament Pràctica 3 mode E-P-S



**Figura 67** Funcionament Pràctica 3 mode U-D-L



## 6.2.4 Pràctica 4

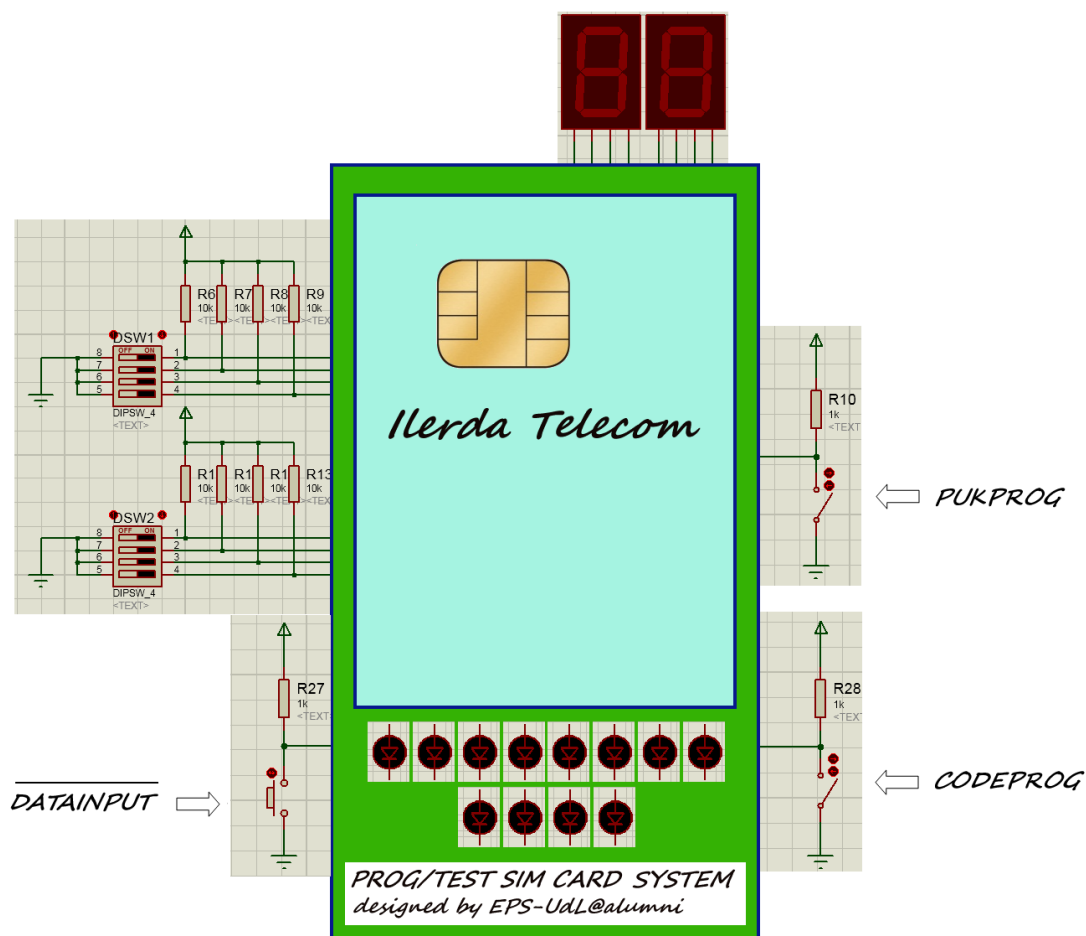
### ➤ Enunciat:

L'objectiu d'aquesta pràctica és dissenyar i simular un circuit digital combinacional + seqüencial de dificultat avançada.

### ➤ Exercici 1

Lleida 2016.- L'empresa Ilerda Telecom és un nou operador de telefonia mòbil que operarà a les Terres de Ponent durant el proper any 2017. S'ha posat en contacte amb l'EPS per tal de desenvolupar conjuntament un sistema de programació i test de targetes SIM per tal de poder oferir el seu servei de telefonia un cop els usuaris de la companyia comprin el darrer smartphone que l'empresa introduirà al mercat a principis de 2017: **Lo marracophone7**.

El diagrama de blocs i d'entrades/sortides del sistema a dissenyar és el següent:



El sistema té els següents modes de funcionament:

Mode funcionament	Mode núm	PUKPROG	CODEPROG
Validar PIN	0	0	0
Programació PIN	1	0	1
Programació PUK	2	1	0
Validar PUK	3	1	1

La seqüència lògica de funcionament del sistema és:

- 1) **1er** cal programar el PIN de la targeta (recordeu que el sistema dissenyat l'ha d'utilitzar un tècnic de l'operador de telefonia mòbil!!!).
- 2) **2on** cal programar el PUK de la targeta.
- 3) **3er** cal validar el PIN.
- 4) **4rt** cal validar el PUK.

Un cop implementades aquestes fases, la targeta ja està llesta per ser lliurada a l'usuari final que l'utilitzarà en el terminal Lo marracophone 7.

Anem a veure com s'ha de comportar el sistema en cadascun dels modes de funcionament:

#### **1er cal programar el PIN de la targeta:**

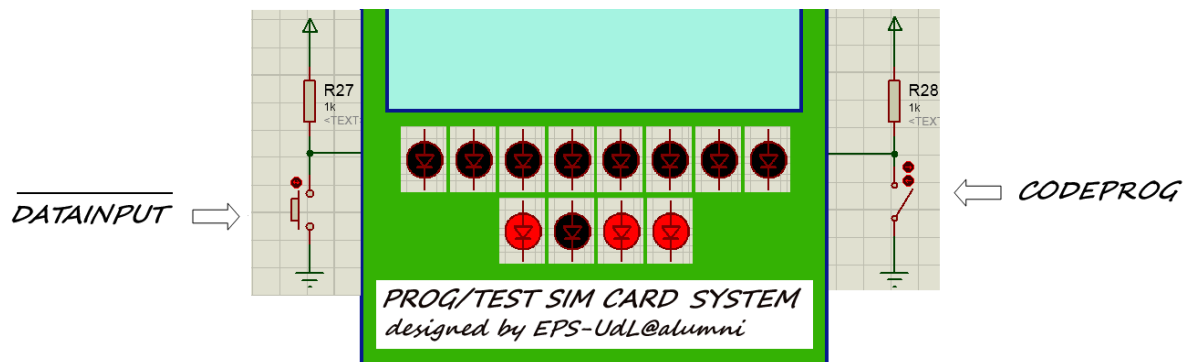
**Mode 1** (Programació PIN, **PUKPROG** = 0 i **CODEPROG** = 1)

Un cop el mode de funcionament ha estat fixat mitjançant els senyals **PUKPROG** i **CODEPROG**, el tècnic de l'operador mòbil que utilitza el sistema dissenyat pot introduir un codi PIN de 4 bits a la memòria ROM del xip.

Per fer-ho, cal que utilitzi qualsevol dels dos DIP switchs (DIP-SW) que té disponibles (de 4 bits cadascun d'ell) i, un cop fixat el codi PIN de 4 bits en el DIP switch, cal que premi el polsador **DATAINPUT** (senyal actiu per nivell baix... és a dir, quan es prem es genera un 0). En el moment que es genera un 0 al senyal DATAINPUT, el codi PUK queda gravat al sistema i es mostra a través d'un grup de 4 LEDS.

És evident que mentre els senyals **PUKPROG** i **CODEPROG** no canviïn, el tècnic pot variar el codi que hi ha al DIP switch, però només es gravarà al sistema (i es mostrarà a través dels LEDS) si es prem el botó associat al senyal **DATAINPUT** quan es vulgui realitzar l'acció de gravar el codi.

Per exemple, si gravem el PIN 1011, el sistema hauria de mostrar el següent:



## 2on cal programar el PUK de la targeta:

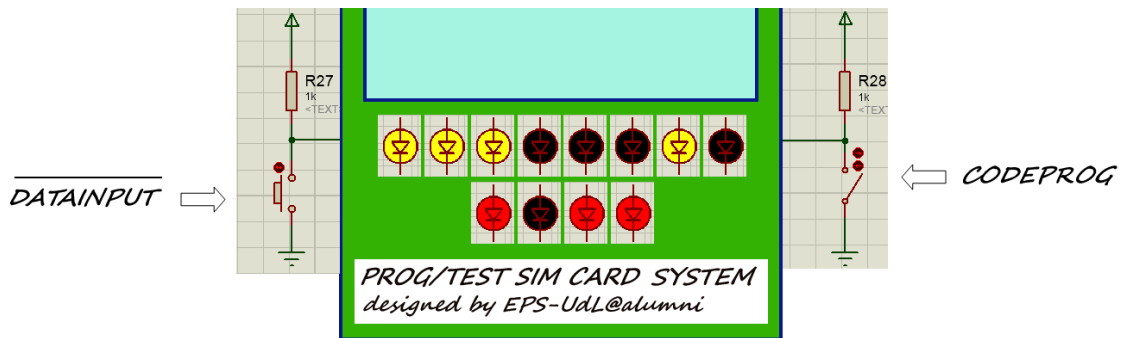
**Mode 2** (Programació PUK, **PUKPROG** = 1 i **CODEPROG** = 0)

Un cop el mode de funcionament ha estat fixat mitjançant els senyals **PUKPROG** i **CODEPROG**, el tècnic de l'operador mòbil que utilitza el sistema dissenyat pot introduir un codi PUK de 8 bits a la memòria ROM del xip.

Per fer-ho, cal que utilitzi els dos DIP switchs (DIP-SW) que té disponibles (de 4 bits cadascun d'ells) i, un cop fixat el codi PUK de 8 bits en el DIP switch, cal que premi el polsador **DATAINPUT** (senyal actiu per nivell baix... és a dir, quan es prem es genera un 0). En el moment que es genera un 0 al senyal **DATAINPUT**, el codi PUK queda gravat al sistema i es mostra a través d'un grup de 8 LEDS.

És evident que mentre els senyals **PUKPROG** i **CODEPROG** no canviïn, el tècnic pot variar el codi que hi ha al DIP switch, però només es gravarà al sistema (i es mostrarà a través dels LEDS) si es prem el botó associat al senyal **DATAINPUT** quan es vulgui realitzar l'acció de gravar el codi.

Per exemple, si gravem el PUK 11100010 el sistema hauria de mostrar el següent:



### 3er cal validar el PIN:

**Mode 0** (Validar PIN, **PUKPROG** = 0 i **CODEPROG** = 0)

Un cop el mode de funcionament ha estat fixat mitjançant els senyals **PUKPROG** i **CODEPROG**, el tècnic de l'operador mòbil que utilitza el sistema dissenyat ha de validar el codi PIN.

Per fer-ho, cal que utilitzi qualsevol dels dos DIP switchs (DIP-SW) que té disponibles (de 4 bits cadascun d'ell) i, un cop fixat el codi PIN de 4 bits en el DIP switch, cal que premi el polsador **DATAINPUT** (senyal actiu per nivell baix... és a dir, quan es prem es genera un 0).

En el moment que es genera un 0 al senyal **DATAINPUT**, el sistema valida el codi PIN introduït amb el codi PIN gravat. Si el codi PIN introduït (que no s'ha de mostrar a través dels LEDs, ja que els LEDs sempre han de mostrar el codi PIN gravat a la targeta) és correcte, llavors el LED de la dreta s'encendrà indicant que el PIN s'ha validat correctament.

### 4rt cal validar el PUK:

**Mode 3** (Validar PUK, **PUKPROG** = 1 i **CODEPROG** = 1)

Un cop el mode de funcionament ha estat fixat mitjançant els senyals **PUKPROG** i **CODEPROG**, el tècnic de l'operador mòbil que utilitza el sistema dissenyat ha de validar el codi PUK.

El tècnic només disposa d'un intent per validar el codi PUK i a més a més ho ha de fer abans que passin 5 segons a partir del moment en que entra al mode 3 (Validar PUK).

Per fer-ho, cal que utilitzi els dos DIP switchs (DIP-SW) que té disponibles (de 4 bits cadascun d'ell) i, un cop fixat el codi PUK de 8 bits en el DIP switch, cal que premi el polsador **DATAINPUT** (senyal actiu per nivell baix... és a dir, quan es prem es genera un 0).

En el moment que es genera un 0 al senyal **DATAINPUT**, el sistema valida el codi PUK introduït amb el codi PUK gravat. Si el codi PUK introduït (que no s'ha de mostrar a través dels LEDs ja que els LEDs sempre han de mostrar el codi PUK gravat a la targeta) és correcte, s'activarà el LED corresponent a la validació del PUK indicant que la validació s'ha realitzat correctament.

➤ **Realització de la pràctica:**

➤ **Carrega de dades:**

Per realitzar la carrega dels 8 bits del puk o dels 4 bits del codi segons el mode de funcionament en el que estiguem, no podem fer-ho únicament amb portes lògiques. En aquests casos fa falta un registre de desplaçament, en aquest cas un registre de desplaçament de carrega en paral·lel.

➤ **Registres de desplaçament:**

Un registre de desplaçament és un circuit digital seqüencial (és a dir, que els valors de les seves sortides depenen de les entrades i dels seus valors anteriors), consistent en una sèrie de biestables. Normalment, aquest biestables són de tipus D, connectats en cascada, que funcionen de manera síncrona amb la mateixa senyal de rellotge. Segons les connexions entre els biestables del registre, es pot aconseguir un desplaçament cap a l'esquerra o cap a la dreta de la informació emmagatzemada.

Cal destacar que desplaçar cap a la dreta un conjunt de bits equival a multiplicar per dos i desplaçar cap a la dreta equival a dividir entre 2.

També existeixen registres bidireccionals, que poden funcionar en ambdós sentits.

Depenent del tipus d'entrades i sortides, els registres de desplaçament es poden classificar com:

- **Sèrie-Sèrie:** Només són accessibles externament l'entrada del primer flip-flop i la sortida de l'últim.
- **Paral·lel-Sèrie:** Són accessibles les entrades de tots els flips-flops, però només la sortida de l'últim. Aquest tipus s'utilitza per convertir dades sèrie en paral·lel i a la inversa.
- **Sèrie-Paral·lel:** Al contrari que l'anterior, són accessibles totes les sortides dels flips-flops, però només la primera entrada.
- **Paral·lel-Paral·lel:** Tant les entrades com les sortides són accessibles. S'utilitzen per càlculs aritmètics.

Aplicacions:

- Generador Pseudoaleatori
- Multiplicador en sèrie
- Registre de aproximacions successives
- Retardador de bits

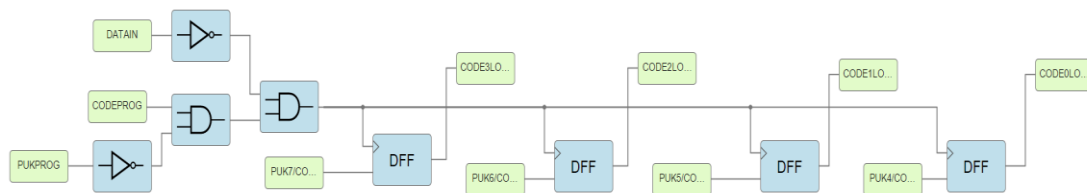
➤ **Introducció Codi PIN:**

Un cop el tècnic de l'operador de mòbil ha fixat mitjançant els senyals PukProg i CodeProg el mode I de funcionament, s'ha d'introduir el codi PIN format per un nombre de 4 bits.

Per poder emmagatzemar el codi PIN s'utilitza un registre format per 4 biestables de tipus D connectats en cascada.

El valor del codi PIN introduït pel tècnic només es quedarà emmagatzemat al registre quan el valor de DATAIN serà de 0. Això provocarà que el codi PUK es quedi gravat al sistema.

Per controlar que el codi es quedi emmagatzemat quan l'operari està en el mode 1, controlem la senyal de rellotge del registre amb una porta AND de 3 entrades, assegurant-nos que només funcionarà aquest registre quan el valor de els senyals PukProg i CodeProg siguin "0,1" respectivament i DATAIN generi un 0.



**Figura 68** Registre de 4 bits

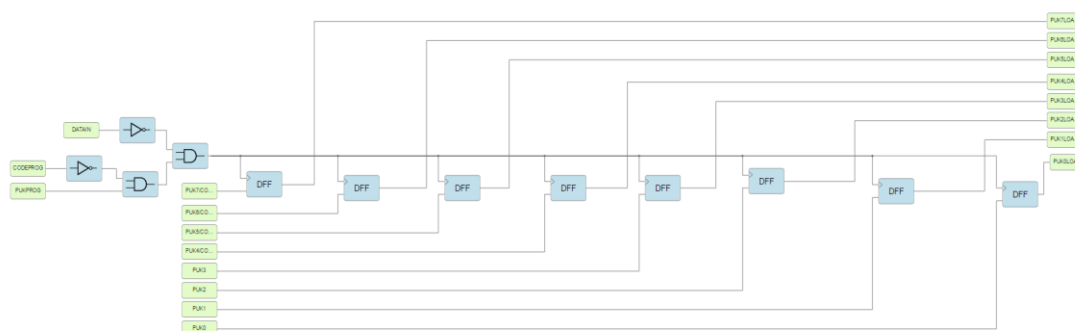
### ➤ Introducció del Puk:

El Puk que volem introduir és un nombre binari que està format per 8 bits, per tant, necessitarem construir un registre per poder emmagatzemar el seu valor quan el operari estigui en Mode 2.

Per realitzar-ho, construirem el registre mitjançant 8 flip-flops connectats en cascada, on el seu valor D per cada flip-flop serà el corresponent dígit en binari del Puk.

Volem que el codi Puk només quedi enregistrat com ens diu l'enunciat quan el valor de PukProg sigui "1" i CodeProg sigui "0". Per tant, mitjançant lògica binària connectem aquestes dos entrades (PukProg i CodeProg) mitjançant una porta AND de 3 entrades.

Quan l'operari premi el polsador de DATAIN, la senyal activarà la porta AND i arribarà un 1 a la senyal de rellotge del registre, de forma que el bit assignat a cada flip-flop quedarà enregistrat.



**Figura 69** Registre de 8 bits

### ➤ Validació del PIN

Un cop l'operari ha establert el codi Puk, és el moment de validar el codi PIN. Per a poder validar-ho, l'operari ha de ficar el dispositiu en Mode 0, mitjançant la introducció de les senyals CodeProg i PukProg, amb un valor de "0" binari les dues.

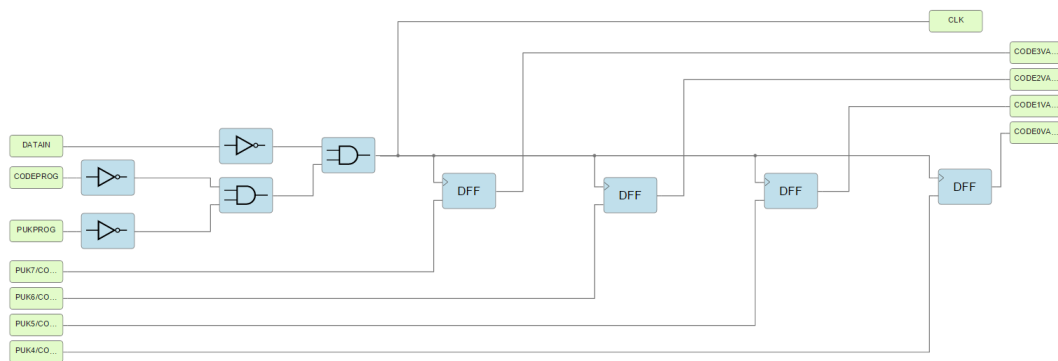
Un cop el dispositiu ja està en Mode 0, l'operari ha d'introduir el codi PIN a validar. Aquest codi PIN s'emmagatzemarà de la mateixa forma que les dos vegades anteriors. Mitjançant un registre format per 4 biestables de tipus D col·locats en cascada amb un senyal de rellotge que només s'activi quan els senyals de control CodeProg i PukProg i DataIn valguin 0.

Un cop l'operari ha carregat el codi PIN a validar, mitjançant un conjunt de portes lògiques tal i com es mostra, es procedeix a la comprovació de que és el mateix nombre.

A la sortida del circuit de portes lògiques hi haurà la senyal CodeOK, que serà de valor alt quan el codi introduït per l'operari sigui vàlid.

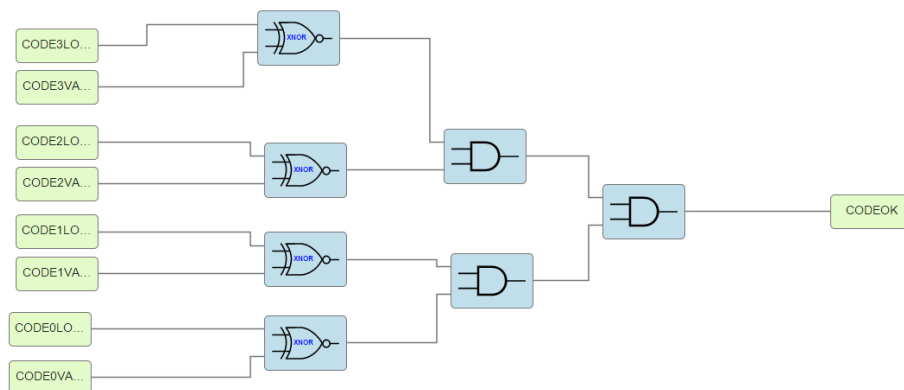
Si la validació del PIN s'ha realitzat correctament, CodeOK valdrà un 1 i s'encendrà el LED.

➤ **Introducció Codi PIN a validar:**



**Figura 70** Registre de 4 bits per validar Codi Pin

➤ **Validació del Codi PIN:**



**Figura 71** Validació del codi Pin



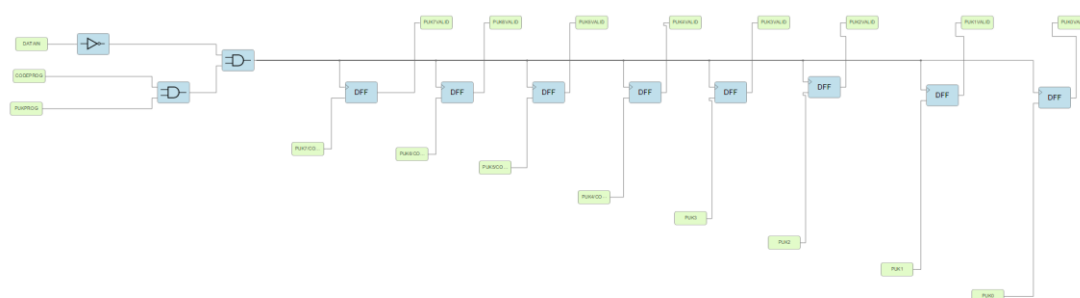
## ➤ Validació del Puk

De la mateixa forma que la validació del PIN, l'operari ha de validar el PUK.

Per poder validar-ho, els senyals de control han d'ésser CodeProg=0 i PukProg=0.

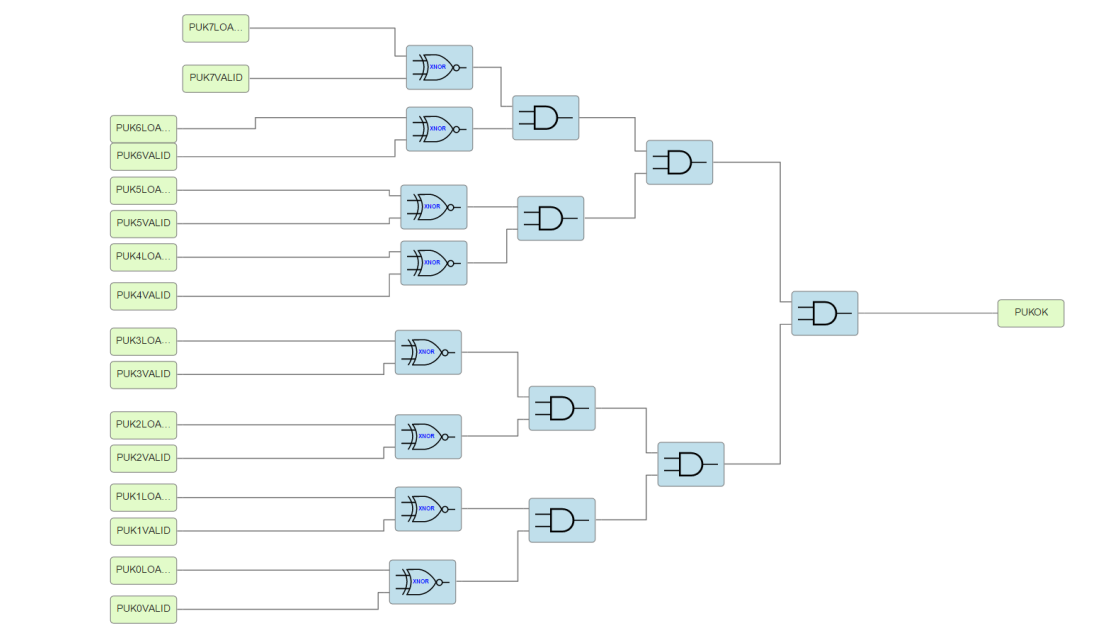
Un cop l'operari té marcat el codi Puk per validar, polsa el botó de DATAIN.

Al pulsar-lo, el senyal del rellotge pertanyent al registre encarregat d'emmagatzemar el Puk a validar s'activa i queda el Puk gravat dins del registre, format per 8 flip-flops tipus D en cascada.



**Figura 72** Registre per validar codi PUK

Un cop ha quedat registrat el codi Puk a comparar, mitjançant un conjunt de portes lògiques es compara cada bit del codi Puk a validar amb l'original i s'obté com a sortida la senyal PukOk, que s'activarà quan el codi a validar sigui correcte.



**Figura 73** Validació del codi PUK

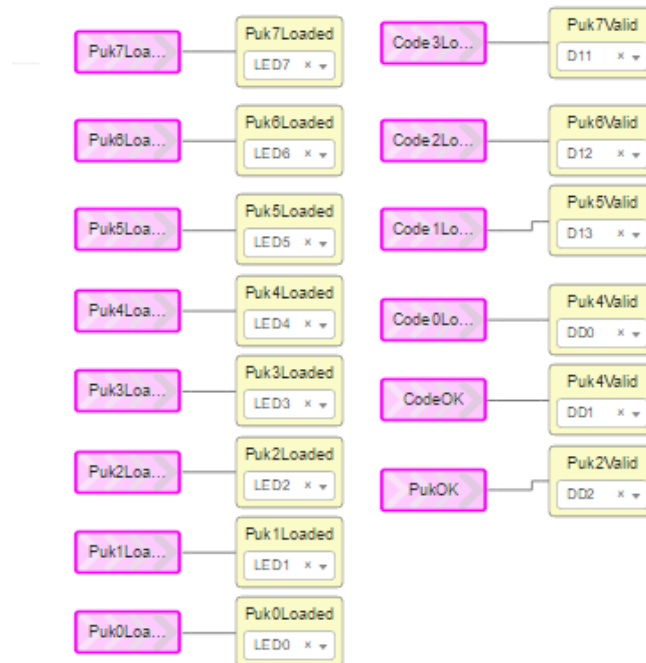
➤ **Connexions a l'IceStudio-LEDS**

Connexions a la placa de training FPGA-UdL

Sortides PUKLoaded [7..0] i CodeLoaded [3..0]

PUK7Loaded	➔	LED7
PUK6Loaded	➔	LED6
PUK5Loaded	➔	LED5
PUK4Loaded	➔	LED4
PUK3Loaded	➔	LED3
PUK2Loaded	➔	LED2
PUK1Loaded	➔	LED1
PUK0Loaded	➔	LED0
Code3Loaded D11	➔	RED_A
Code2Loaded D12	➔	YELLOW_A
Code1Loaded D13	➔	GREEN_A
Code0Loaded DD0 (A0)	➔	RED_B
CodeOK DD1 (A1)	➔	GREEN_B
PukOK DD2 (A2)	➔	YELLOW_B

**Taula 12** Connexions a la training board



**Figura 74** Connexions IceStudio-Training Board

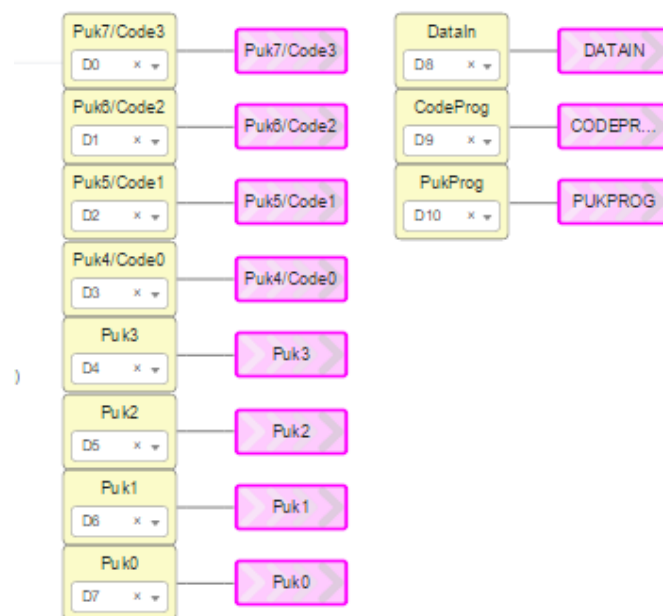
### ➤ Connexions IceStudio-Entrades

Connexions a la placa de training FPGA-UdL

Entrades PUK [7..0] i Code [3..0]

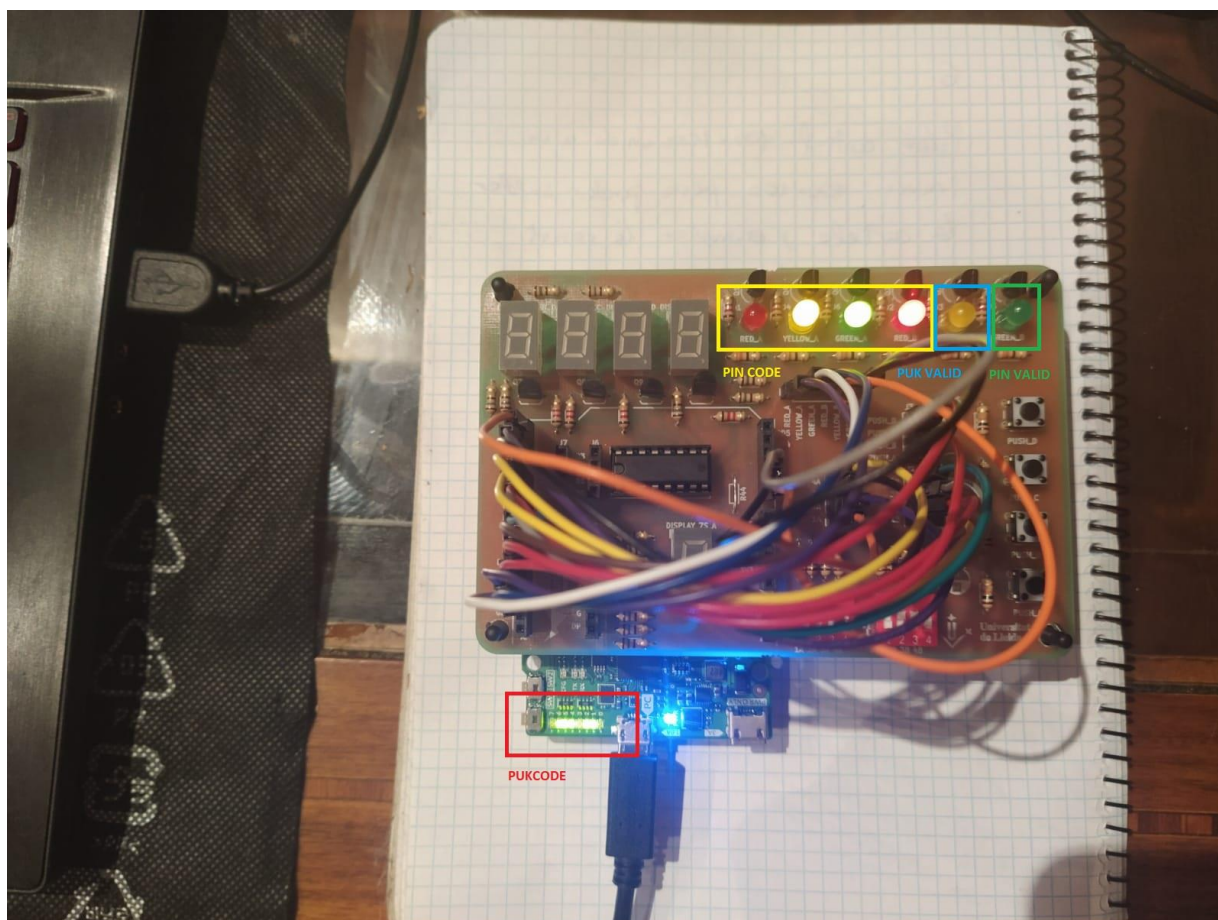
PUK7/Code3	D0 <-- 1A
PUK6/Code2	D1 <-- 2A
PUK5/Code1	D2 <-- 3A
PUK4/Code0	D3 <-- 4A
PUK3	D4 <-- 1B
PUK2	D5 <-- 2B
PUK1	D6 <-- 3B
PUK0	D7 <-- 4B
DATAIN	D8 <-- PUSH_D (cal prémer per fer un flanc de pujada amb DATAIN)
CODEPROG	D9 <-- PUSH_C (quan es prem es genera un 0)
PUKPROG	D10 <-- PUSH_B (quan es prem es genera un 0)

**Taula 12** Connexions Entrades-TrainingBoard



**Figura 75** Connexions IceStudio-Training Board

➤ **Funcionament Pràctica 4:**



**Figura 76** Funcionament Pràctica 4

## 7 . TinyFPGA

Tal com s'ha explicat anteriorment sobre la training board de la UdL, el nucli és la placa d'entrenament Icezum Alhambra II.

Actualment, el laboratori d'electrònica de la UdL disposa de 24 training board.

Però, tenint en compte que al laboratori solament es disposen de 12 plaques Icezum Alhambra per a connectar a la training board, en realitat únicament es poden utilitzar la meitat. És a dir, que comptem amb 12 training board complertes.

Durant les sessions de pràctiques s'acostuma a utilitzar una training board cada dos alumnes, però degut a les mesures de seguretat establertes per a combatre l'actual pandèmia COVID19, és molt probable que no es pugui continuar amb aquesta pràctica d'una training board per cada parella d'alumnes. D'altra banda, en el cas que algun any el nombre d'alumnes matriculats fos superior al que és habitual, això implicaria la insuficiència de training boards disponibles per a tots els alumnes.

És per això que l'última part d'aquest treball ha consistit en el disseny d'un PCB que actuï com a "socal" i es pugui incorporar la tinyFPGA mitjançant connectors soldats. D'aquesta forma, es podria connectar el "socal" a la training board per així poder fer-les funcionals.

Cal tenir en compte que degut a que el preu de la TinyFPGA és considerablement inferior al de l'Icezum Alhambra II, amb aquest disseny també es podria arribar a reemplaçar les FPGA Alhambra II de la training board per TinyFPGAs.

El preu de la tinyFPGA és de 34.2+IVA mentre que el de l'Alhambra II és de 49.9+IVA.

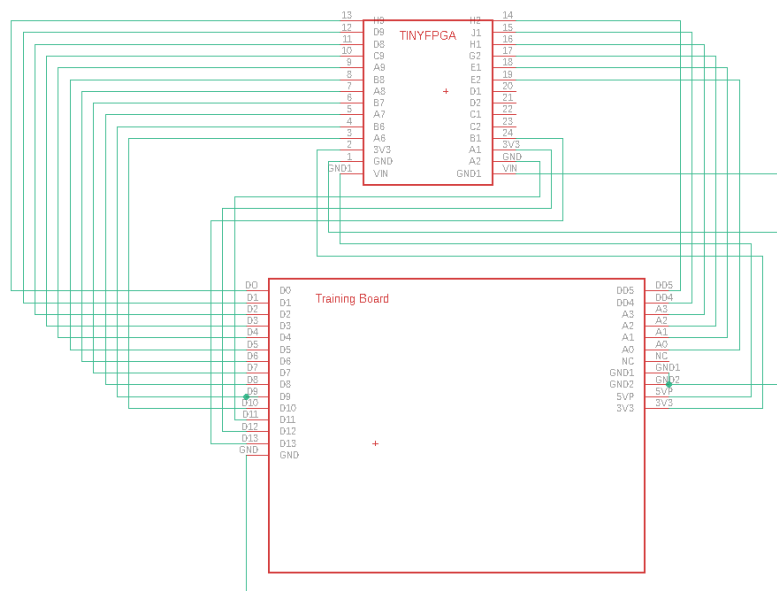
El disseny de l'adaptació per incorporar la tinyFPGA s'ha realitzar mitjançant l' EAGLE. Per a fer els schematics s'ha dissenyat dos components per a poder connectar els pins de la tinyFPGA amb els de la training board.

Coincidint amb el fet que els pins d'E/S de la training board són iguals que els d'Arduino Uno, s'ha dissenyat un component amb els pins col·locats per a que coincideixin amb la training board.

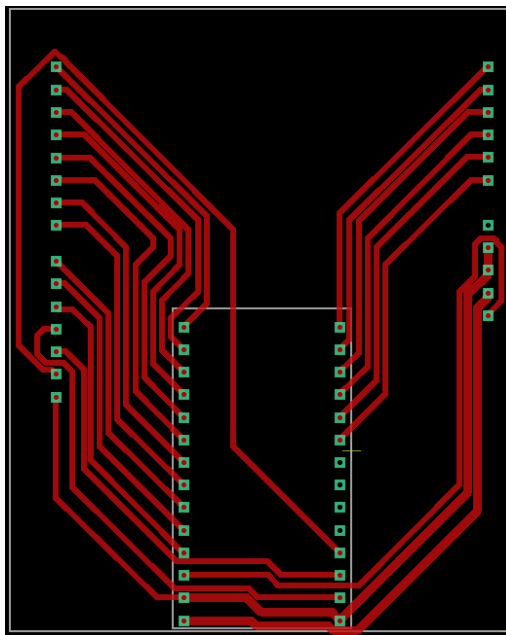
En el disseny del PCB s'han tingut en compte diverses consideracions:

- Les pistes tindran una amplada de 0.6 mm per les pistes que porten senyals i de 0.9 mm per les pistes connectades a alimentació o terra.
- El disseny s'ha fet en una sola capa.
- Evitar pistes que formin angles de 90° per evitar problemes de transmissió.

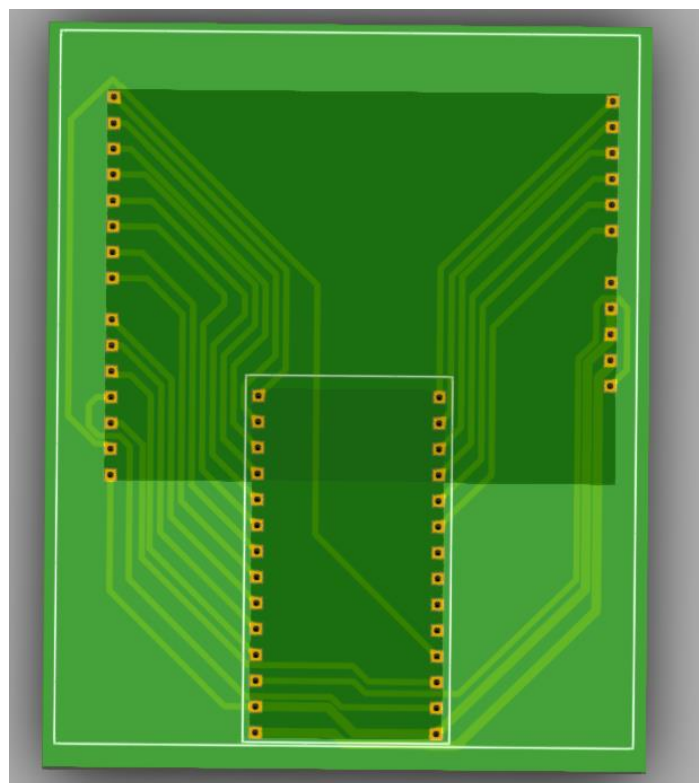
## 7.1 Schematic



## 7.2 PCB



**Figura 78** Disseny PCB



**Figura 79** Disseny PCB en 3D

## 8. CONCLUSIÓ

La formació pràctica de l'estudiant dels graus d'enginyeria ha d'ésser dinàmica seguint els canvis tecnològics, socials i industrials actuals.

La proposta d'aquest treball de fi de grau parteix de l'objectiu de complementar les pràctiques de laboratori tradicionals que els estudiants realitzaven en les assignatures d'electrònica digital mitjançant l'ús de FPGAs.

Mitjançant la plataforma de desenvolupament de baix cost implementada en la UdL, s'ha pretès que l'estudiant compti amb eines complementàries perquè puguin realitzar un major nombre de pràctiques d'una forma més autònoma.

Amb aquest TFG s'ha pogut afegir una nova eina, un curs d'aprenentatge en línia d'introducció a l'electrònica digital mitjançant el qual es posa a disposició dels alumnes un ampli volum d'informació i pràctiques, que permeti a l'estudiant realitzar activament pràctiques d'electrònica digital de manera fàcil i senzilla, mitjançant l'ús de Hardware i Software lliure per programar les FPGAs.

El gran èxit de les FPGAs i el seu principal avantatge respecte a dispositius similars és que es poden reprogramar, la qual cosa afegeix una enorme flexibilitat dels seus dissenys. D'altra banda, l'impacte que està tenint el moviment maker i l'aposta per l'ús d'eines de codi obert, HW/SF lliure, pot tenir un considerable avantatge a l'hora de realitzar les pràctiques i millora els processos d'aprenentatge dels estudiants, ja que els costos de desenvolupament i adquisició són molt menors i el temps emprat en el procés de disseny i desenvolupament és també menor.

Hem pogut observar en la pandèmia de COVID19 que el moviment maker s'ha bolcat en la fabricació de viseres de protecció o el disseny de respiradors mitjançant impressió 3D. I és clar que aquest moviment ha vingut per a quedar-se.

La Universitat ha de fomentar noves propostes per facilitar la transformació de la metodologia tradicional d'acord amb les tendències actuals, de forma que s'aconsegueixin les competències que la societat exigeix als enginyers.



## 9. REFERÈNCIES BIBLIOGRÀFIQUES

### PLAQUES FPGA

<https://www.profesaulosuna.com/data/files/ELECTRONICA/ELECTRONICA%20DIGITAL%20II/FPGA%201.pdf>

[http://dea.unsj.edu.ar/sisdig2/Field%20Programmable%20Gate%20Arrays\\_A.pdf](http://dea.unsj.edu.ar/sisdig2/Field%20Programmable%20Gate%20Arrays_A.pdf)

<https://www.luisllamas.es/que-es-una-fpga/>

[https://en.wikipedia.org/wiki/Xilinx#Virtex\\_family](https://en.wikipedia.org/wiki/Xilinx#Virtex_family)

<https://www.intel.com/content/www/us/en/products/programmable/fpga.html>

[https://en.wikipedia.org/wiki/ICE\\_\(FPGA\)#Architecture](https://en.wikipedia.org/wiki/ICE_(FPGA)#Architecture)

<https://www.latticesemi.com/Products/FPGAandCPLD/CrossLink>

[https://www.semiconductorstore.com/pages/promo\\_landing/2015/lattice\\_mach/lattice\\_mach.asp](https://www.semiconductorstore.com/pages/promo_landing/2015/lattice_mach/lattice_mach.asp)

<https://www.mouser.es/new/microsemi/microsemi-polarfire-fpga/>

<https://www.microsemi.com/product-directory/fpgas/1688-igloo2>

<https://www.microsemi.com/product-directory/fpgas/1690-proasic3>

<https://www.microsemi.com/product-directory/fpgas/1689-igloo>

<https://www.microsemi.com/product-directory/fpgas/1691-fusion>

<https://www.wired.com/2016/09/microsoft-bets-future-chip-reprogram-fly/>

<https://vhdl.es/fpga/>

<https://github.com/Obijuan/open-fpga-verilog-tutorial/wiki>

[https://www.researchgate.net/publication/268253760\\_GUIA\\_PARA\\_PROGRAMACION\\_DE\\_FPGA](https://www.researchgate.net/publication/268253760_GUIA_PARA_PROGRAMACION_DE_FPGA)

### HW LLIURE/ SW LLIURE

[http://www.translatoruser-int.com/translate?&from=en&to=es&csId=425d794c-d1a6-459e-a4b5-c13ca89fe051&usId=dde0d284-b4f4-4304-9a79-c7dabe24892c&dl=es&ref=SERP\\_ct&dt=2020%2f6%2f15%2022%3a6&h=hHagSwQbcLXZ46gkn6FMDFNJ9EZJFWyF&a=http%3a%2f%2fwww.clifford.at%2ficastorm%2f](http://www.translatoruser-int.com/translate?&from=en&to=es&csId=425d794c-d1a6-459e-a4b5-c13ca89fe051&usId=dde0d284-b4f4-4304-9a79-c7dabe24892c&dl=es&ref=SERP_ct&dt=2020%2f6%2f15%2022%3a6&h=hHagSwQbcLXZ46gkn6FMDFNJ9EZJFWyF&a=http%3a%2f%2fwww.clifford.at%2ficastorm%2f)

[https://es.slideshare.net/obijuan\\_cube/fpgas-libres](https://es.slideshare.net/obijuan_cube/fpgas-libres)

<http://www.clifford.at/icastorm/>

<http://fpgalibre.sourceforge.net/>

<https://www.translatetheweb.com/?from=en&to=es&ref=SERP&dl=es&rr=UC&a=https%3a%2f%2ficastudio.io%2f>

[https://www.translatetheweb.com/?from=en&to=es&ref=SERP\\_ct&dl=es&rr=HE&a=https%3a%2f%2fgithub.com%2fObijuan%2fdigital-electronics-with-open-FPGAs-tutorial%2fwiki%2fV%25C3%25ADdeo-3%3a-La-Icezum-Alhambra-y-otras-placas-con-FPGAs-libres](https://www.translatetheweb.com/?from=en&to=es&ref=SERP_ct&dl=es&rr=HE&a=https%3a%2f%2fgithub.com%2fObijuan%2fdigital-electronics-with-open-FPGAs-tutorial%2fwiki%2fV%25C3%25ADdeo-3%3a-La-Icezum-Alhambra-y-otras-placas-con-FPGAs-libres)

<https://github.com/Obijuan/Curso-Electronica-Digital-para-makers-con-FPGAs-Libres/wiki>

<http://www.learobotics.com/personal/juan/docencia/apuntes-ssdd-0.3.7.pdf>

[https://www.translatetheweb.com/?from=en&to=es&ref=SERP\\_ct&dl=es&rr=HE&a=https%3a%2f%2fgithub.com%2fObijuan%2fopen-fpga-verilog-tutorial%2fwiki](https://www.translatetheweb.com/?from=en&to=es&ref=SERP_ct&dl=es&rr=HE&a=https%3a%2f%2fgithub.com%2fObijuan%2fopen-fpga-verilog-tutorial%2fwiki)

<http://www.clifford.at>

## **ICEZUM ALHAMBRA II / TRAINING BOARD UdL**

<https://github.com/FPGAwards/Alhambra-II-FPGA.wiki.git>

[https://github.com/FPGAwards/Alhambra-II-FPGA/blob/master/doc/pinout/Alhambra%20II%20V1.0A%20-%20Pinout\\_v1.0\\_rev%202.svg](https://github.com/FPGAwards/Alhambra-II-FPGA/blob/master/doc/pinout/Alhambra%20II%20V1.0A%20-%20Pinout_v1.0_rev%202.svg)

A. Saiz- Vela, “Plataforma de Desarrollo de bajo coste para implementar circuitos digitales en FPGAs mediante hardware i software libre” ; Escola Politècnica Superior, Universitat de Lleida (UdL)